Identifying On-Site Users for Social Events: Mobility, Content, and Social Relationship

Zhiwen Yu, *Senior Member, IEEE*, Fei Yi, Qin Lv, and Bin Guo, *Senior Member, IEEE*

Abstract—The wide spread use of social network services, especially location based services, has transformed social networks into an important information source of real-world events. Many event detection systems using geo-tagged posts from social networks have been developed in recent years. Besides detecting real-world events, it is also desirable for government officials, news media, and police, etc., to identify on-site users of an event, from whom we could gather valuable information regarding the process of events and investigate suspects when an event is associated with crime or terrorist. However, due to the high uncertainty of human mobility patterns and the low probability of users sharing their location information, it is difficult to identify on-site users while a social event unfolds, and research work in this area is still in its infancy. In this paper, we propose a Fused fEature Gaussian prOcess Regression (FEGOR) model, which exploits three influential factors in social networks for on-site user identification: *mobility influence*, *content similarity*, and *social relationship*. By integrating these factors, we are able to estimate the distance between a user and a social event even when the user's location profile is unknown, thus identify on-site users. Experiments on a real-world Twitter dataset demonstrate the effectiveness of our model, achieving a minimum mean absolute error of 1.7km and outperforming state-of-the-art methods.

Index Terms—On-site user, user-social event distance, mobility influence, content similarity, social relationship, gaussian process regression

1 Introduction

In the age of social network services, where Twitter[1] changed its prompt from "*What are you doing*" to "*What's happening*" in 2009 and Mark Zuckerberg made it clear that he wanted Facebook[2] to serve as the "*World's Newspaper*", the way we consume information and the types of data we generate have changed significantly. According to surveys by Pew Research Center[3], in 2012, nearly 49 percent U.S. adults received news on social networks instead of traditional newspapers, and that number increased to 62 percent by 2016 [1]. Social network users are more likely to post newsworthy materials such as on-site photos, videos, or other types of data about social events, instead of only text-based status from themselves. Such revolutionary changes have led us into a new world which can make great use of social and community intelligence [30]. Especially with the development of location based services, geo-tagged posts (e.g., user tweets associated with GPS coordinates) from social networks users offer rich information about our society, and have inspired a line of works aiming at real-world event detection [15], [16], [17], intelligent location based systems [27], [28], [29], crowd sourced data collection [57], [58], [59], and other applications, which can be of great importance to capture city dynamics for social goods.

In recent years, some serious events such as the "2013 Boston Marathon bombings"[4], the "2014 Shanghai stampede"[5] and the "2015 Baltimore protests"[6] have caused a lot of riots that threatened public safety. This has motivated many works on automatic and efficient event detection using social networks to ensure social safety. Apart from detecting real-world events from social networks, government, news media, and the police have discovered the potential value of identifying on-site users among the large crowd of users who are reporting social events. Specifically, on-site users are witnesses of evolving social events and can provide useful information to different organizations. For instance, news media could obtain meaningful information from on-site users to help understand the progression of social events, and the police could investigate suspects using on-site users' testimonies when an event is associated with crime or terrorists. In particular, the "2013 Boston Marathon bombings" triggered a huge social panic, and it was the photos and contents posted by witnesses on social networks that helped police track down the suspects.

To identify on-site users, the traditional and straight forward way is to search users whose tweets are geo-tagged and located within a certain range (e.g., 200 meters) of the center location of a target social event. However, due to personal preference and privacy concerns, many social media users, whether consciously or unconsciously, may not expose their real location, especially when tweeting about

1. https://www.twitter.com/
2. https://www.facebook.com/
3. http://www.pewresearch.org/

- *Z. Yu, F. Yi, and B. Guo are with the School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China. E-mail: {zhiwenyu, guob}@nwpu.edu.cn, yifeinwpu@gmail.com.*
- *Q. Lv is with the Department of Computer Science, University of Colorado Boulder, Colorado, CO 80309-0430. E-mail: qin.lv@colorado.edu.*

Manuscript received 26 Dec. 2016; revised 27 Sept. 2017; accepted 8 Jan. 2018. Date of publication 18 Jan. 2018; date of current version 2 Aug. 2018. (Corresponding author: Zhiwen Yu.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2794981

4. https://en.wikipedia.org/wiki/Boston_Marathon_bombing
5. https://en.wikipedia.org/wiki/2014_Shanghai_stampede
6. https://en.wikipedia.org/wiki/2015_Baltimore_protests

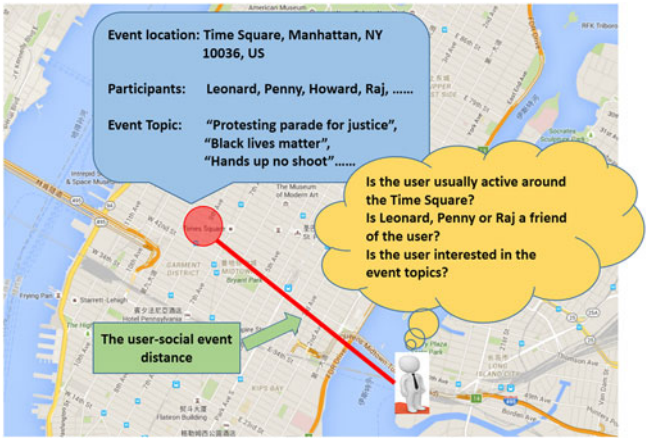


Fig. 1. Illustration of determining *user-social event distance* for on-site user identification. Intuitively, a real-world event occurs at a specific location and is associated with its participants and topics. If a user's mobility pattern is similar to the event participants' mobility patterns, the user is interested in the event's topic, and many of the users' friends are near the event, we could infer that the user is more likely to be active around the event location than those who do not have these features.

some social events. It is reported that only 34 percent of Twitter users have meaningful location information in their profiles and less than 1 percent of Twitter users tag their tweets with GPS location [2], [3]. As such, searching by explicit user location information would return a much smaller set of users than the actual on-site users, which maybe insufficient to capture the overall and detailed information of a social event. The sheer volume of social media users and their posts and the lack of user location information make it particularly challenging to identify on-site users. Therefore, an intelligent system that can automatically identify adequate on-site users would prove to be invaluable for various usage scenarios. This is precisely what we would like to accomplish in this work.

User location information is usually private and can be highly sensitive. According to the location privacy issue discussed in [4], it is possible to achieve the goal of identifying on-site users without compromising user privacy. In this paper, instead of determining the absolute location of each user, we aim to estimate the *relative location*, i.e., the distance between the user and the event, which we refer to as the *User-Social Event Distance*. Specifically, our model is able to compute how far away a user is from the target social event. A smaller user-social event distance indicates a higher probability that the user is on-site, without telling us the actual location of the user. Fig. 1 gives a more comprehensive illustration of our model.

Previous studies [5], [6], [7] have emphasized that the motivation of user participate in social events can be classified into different factors: mobility, content, and social relationship. In this work, we explore the following three features that can be helpful to estimate the *User-Social Event Distance*: *1) Mobility Influence*. This is one key factor that determines whether a user will attend or be interested in a target social event. This feature measures the historical trajectory similarity between a specific user and all participants of an event. Intuitively, if a user has similar trajectories as event participants, it could be inferred that the user would attend or be close to the same event. *2) Content Similarity*. This feature quantifies the textual similarity between a user's tweets and event topic. Each social event can usually be associated with a specific theme that describes the event topic. If a user is near the event, he/she may tweet more about the event, leading to a higher content similarity than those who are far away from the event. *3) Social Relationship*. A user's social relationship also plays an important role when it comes to attending a social event. If a user's friends are among the ones attending the event, then it is more likely that the user would attend the same social event, compared with other users whose friends are not on the list of event participants. A more detailed explanation of these three features can be found in the following sections.

Based on the extracted features above, we then propose a **F**used f**E**ature **G**aussian pr**O**cess **R**egression (*FEGOR*) model, which formulates and parameterizes the three features into an integrate function for *User-Social Event Distance* estimation. Our work makes the following contributions.

- We propose a **F**used f**E**ature **G**aussian pr**O**cess **R**egression (*FEGOR*) model that combines *Mobility Influence*, *Content Similarity*, and *Social Relationship* to estimate the relative location of users w.r.t events, named as *User-Social Event Distance*, which is used to identify on-site users.
- Our method transforms all absolute GPS locations into relative distances between users and social events. Based on such location projection and transformation, we not only accomplish the goal of identifying on-site users for social events, but also protect the individual location privacy in a coarse-grained level.
- We conduct real-world experiments using data collected through the TwitterAPI [7] between Nov. 1, 2014 and Jan. 13, 2015, which consists of 6,727 users and 1,187,847 tweets. The experiments demonstrate the effectiveness and efficiency of our proposed solution.

The remainder of this paper is organized as follows. In Section 2, we review the related work. The framework of our proposed model is illustrated in Section 3 along with the problem definition. In Section 4, we present the feature modeling methods, both for social events and users, as well as the functions for computing the similarity between distinctive features. We then propose the inference model for parameter learning and user-social event distance estimation in Section 5. We present experimental results in Section 6. Finally, we conclude our work and discuss possible applications in Section 7.

2 Related Work

Our work is most relevant to two lines of research as follows: *Locating Social Media Users* and *Understanding User-Event Relationship*.

2.1 Locating Social Network Users

Knowing user locations can be valuable for many application scenarios. Although smart phones and other mobile devices are capable of recording the location information of social media users, most users choose not to expose their locations. Therefore, a branch of work tried to estimate the location profile of users based on several types of

7. https://github.com/geduldig/TwitterAPI

information. Cheng et al. [11] first proposed to infer a Twitter user's city-level location based purely on the content of the user's tweets. They discovered that a user's tweets may encode some location-specific content: either specific place names or certain words or phrases are more likely to be associated with certain locations than others. Ryoo et al. [11] further extended such concept, in which they applied textual content to infer the location of social media users in South Korea and were able to identify 60 percent of users within 10 km of their real locations [12].

Yamaguchi et al. [13] took local events into consideration when estimating the location profiles of social media users. They proposed an online location inference method over social streams that exploited the spatiotemporal correlation. Similarly, Dalvi et al. [10] studied the geographic aspects of tweets and proposed a user-level model for spatial encoding in tweets. Sadilek et al. [14] attempted another way to locate users, which explored the interplay between people's locations, interactions, and their social ties within a large real-world dataset. Tarasov et al. [18] also applied social interaction between users to improve the performance of predicting user locations. Wang et al. [55] proposed a gradient descent based approach to locate the user's moving destination. Apart from those existing studies, our work further associates user locations with social events to estimate a user's location.

2.2 Understanding User-Event Relationship

Apart from inferring user locations, it would be helpful to understand the relationship between users and social events. Yi et al. [56] have emphasized the effectiveness of associating different aspects of data source for understanding social dynamics. Rong et al. [6] identified three factors: social, external and intrinsic influence, which can be utilized to explain the emergence of specific events. Besides, there is a line of works that aim to detect social events. Pan et al. [15] addressed the problem of detecting and describing social event (traffic anomalies) using crowd sensing data. Wang et al. [16] utilized traffic surveillance cameras and social media to detect events. And Sakaki et al. [17] developed a probabilistic spatiotemporal model to monitor tweets and to detect a target event (e.g., earthquake).

Following the research on understanding the dynamics of social events and estimating the locations of target social events, other researchers attempted to discover the relationship between users and social events. Georgiev et al. [7] applied data from Foursquare[8] to analyze event patterns. They evaluated several hypotheses on the motivating factors of user participation and confirmed that social aspects play a key role in determining the likelihood of users participating in an event. Their work also emphasized that the combination of temporal, spatial, and social aspects can be more powerful in understanding users' interest in social events. Du et al. [5] exploited individual behavior patterns in Event-Based-Social-Networks (EBSNs). They aimed to predict activity attendance and discovered a set of factors: content preference, context (spatial and temporal) and social influence, which affect individual's attendance of activities (events) in EBSNs. They further proposed a Singular Value Decomposition with Multi-Factor Neighborhood (SVD-MFN) algorithm to predict activity attendance for each user.

In this work, we focus on identifying on-site users w.r.t real-world events, which is an extension to the research areas mentioned above. Different from those existing works, we attempt to estimate relative locations of social network users when specific real-world events occur. We propose to extract three factors: mobility influence, content similarity, and social relationship, to help build a system for *User-Social Event Distance* estimation. The following sections explain our methods in detail, and the experimental results show the effectiveness of our work as compared with prior works.

3 Problem Statement and System Overview

In this section, we first give the formal definitions of social events and on-site users in our work, then we formulate our problem and give a systematic overview of our proposed solution.

3.1 Social Event and On-Site User

Social Event is a collection of space/time regions containing a group of participants. A social event can be initiated, organized, and executed by some of the participants for certain purpose or it can be accidents that occur randomly. For example, music festivals, traffic jams, parades and crime violence are all social events. Generally, social events attract a lot of attention from real-world users, and understanding the relationship between events and its participants can be valuable and help capture the dynamics of our society. Specifically, we take parade protest as case study to evaluate the proposed method in our work.

On-site User is a person who witnesses social events, and can be either a participant or someone passing by. Typically, few of these users would expose their location when they choose to post information about an event on social media. Based on this observation, we identify three types of on-site users: *1) Active Users* are those who post and share what they witnessed on social media along with their location information when a target social event happens; *2) Normal Users* are those who post and share what they witnessed without exposing their location information; and *3) Inactive Users* are those who post nothing on social media. *Active Users* are easy to identify since they expose their locations, while *Inactive Users* are impossible to find since they post nothing on social media. Compared with *Active Users*, there are many more *Normal Users* who post valuable information about social events while hiding their locations. Therefore, in this paper, our goal is to identify the relative location information of *Normal Users*, which helps to capture the dynamics of social events.

3.2 Problem Statement

For each social event E that occurs at location L during a time window T, there is usually a group of event-related users $U_{all}(u_1, u_2, \ldots, u_n)$ who post related information about the event, including active users U_{act} and normal users U_{nor} as we discussed above. However, location information can be sensitive considering users' privacy. Therefore, we transform all absolute GPS data into relative distance measurements both in feature modeling and result deduction for

8. https://foursquare.com/

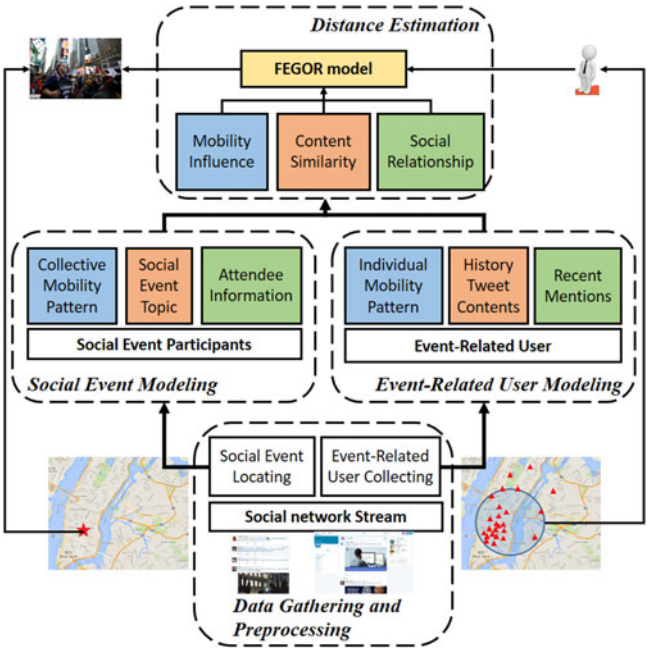


Fig. 2. Identifying on-site users for social events: System overview.

coarse-grained location privacy protection. And the distance measurements between each user u_i in U_{all} and the event E is defined as $Dis(u_i, E)$. The goal in our paper is to estimate $Dis(u_i, E)$ to help identify on-site users, and the assumption is that a smaller $Dis(u_i, E)$ indicates a higher possibility that u_i is on-site. Therefore, the key problem in this paper is how to estimate $Dis(u_i, E)$ accurately and efficiently.

Specifically, a social event E can be represented as a triple $E < \Omega, \Theta, \Phi >$, where Ω is the set of participants, Θ is the topic and Φ is the collaborative mobility pattern extracted from Ω. Each user u_i in U_{all} is also modeled by a triple $u < \omega, \theta, \phi >$, where ω is the recent mentions in user's tweets that indicates his/her social relationship, θ contains the historical tweets' topic from the user and ϕ represents the user's individual mobility pattern. According to these two definitions, we then extract three features: Mobility Influence, Content Similarity and Social Relationship as F_{MI}, F_{CS} and F_{SR} between user and event respectively. Suppose there is a similarity function $Sim(x, y)$ that measures the similarity between two data format x and y. Therefore, we have $F_{MI} \propto Sim(\Phi, \phi)$, $F_{CS} \propto Sim(\Theta, \theta)$ and $F_{SR} \propto Sim(\Omega, \omega)$. And our goal is to estimate $Dis(u, E)$, which is finally formulated as

$$Dis(u, E) = GPR(F_{MI}, F_{CS}, F_{SR}) \quad (1)$$

based on a Gaussian Process Regression (GPR) model. In this function, we assume that a higher similarity between user and the target event across all the three features indicates a smaller distance between user and the event, thus leading to a higher possibility that the user could be on-site.

3.3 System Overview

Fig. 2 gives an overview of our proposed framework, which consists of four components: *Data collection and preprocessing* (bottom), *Social event modeling* (left), *Event-related user modeling* (right), and *User-social event distance estimation* (top).

Data Collection and Preprocessing. Our solution leverages human-generated data from social networks, such as Twitter. Thus, the first step of the framework is data collection and preprocessing, which consists of two main functions: Social Event Localization and Event-related User Collection. Given the stream of user-generated social network data, we use a moving time window (e.g., each hour) to discover hot words and use the method proposed in [11] to locate social events related to the hot words. In the meantime, people who have tweeted about the target social event are identified as event-related users. This process results in two types of information: target social events (with event location) and event-related users, including active users and normal users as we discussed above, which can be used in the following components.

Social Event Modeling. Given a social event identified from the previous step along with its corresponding time window and event location, we identify the event participants as a subset of active users who have tweeted about the event and their locations are within certain range (e.g., 200 meters) of the event center. Using this list of known participants, we can extract features of the event from their historical locations, tweets, and user profiles, resulting in Collective Mobility Patterns (Φ), Social Event Topic (Θ), and Attendee Information (Ω) respectively.

Event-related User Modeling. Event-related users are those who have tweeted about the target social event, which indicates that these users are interested in, and potentially participate in, the specific social event. For each event-related user whose location is unknown during the event time window (i.e., normal users), we would like to estimate the distance between these users and the social event. As shown in Fig. 2, the red star on the bottom-left map indicates the location of the target social event, and the red triangles on the bottom-right map represent the locations of the corresponding active users. We can see that people who tweeted about the same event may be at different distances from the event. For each event-related user, we extract three features including individual mobility pattern (ϕ), historical tweets topic (θ), and recent mentions (ω).

User-Social Event Distance Estimation. Using the three distinct features extracted from both the target social event and its event-related users, we calculate the three factors that may contribute to estimating the user-social event distance: mobility influence (F_{MI}), content similarity (F_{CS}) and social relationship (F_{SR}). We then combine these three factors using the proposed FEGOR model to estimate *User-Social Event Distance*, along with an information entropy based genetic algorithm for parameter learning to identify the weights of different features. Intuitively, we could learn the relationship between these factors and *User-Social Event Distance* using active users, since they all expose their locations that provide us with ground truth to train the estimation model (FEGOR); and then apply this model to infer *User-Social Event Distance* for normal users. However, normal users do not have ground truth (actual distance) for us to evaluate the results. Thus, our experiments in this work is conducted using active users, we divide these users into separate groups: training group and testing group, and in the testing group, we remove distance information (their locations) manually and infer them using our proposed FEGOR model for evaluation.

4 Feature Modeling

In this section, we give explanations on why we propose and how we model the features in our work.

4.1 Mobility Influence

It is intuitive that when a social event happens, users who live close to the event center or usually visit nearby places have higher probabilities to be on-site. However, music concerts, protests, parades, and holiday celebrations usually draw users from separate places far away from their locations. For instance, people who gather at New York Time Square on the New Year's Eve are from different parts of the New York City and even around the world. Thus, it is more important to model the mobility similarity between ordinary users and participants of the event instead of only discovering local people w.r.t a social event. Furthermore, since local people usually participate in nearby events, our participants' mobility pattern also contains local people's mobility pattern. And finally, if a user's mobility is similar to an event's known participants' mobility, we could infer that the user is likely to be on-site.

In specific, we propose collaborative mobility pattern Φ for each social event based on its participants' historical trajectories instead of a static location profile to model its mobility pattern, and individual mobility pattern ϕ for each user. Specifically, in order to protect the location privacy of event-related users, we first transform all their historical absolute GPS locations into a distance value relative to the event center. For example, the historical geo-tagged tweets of a user contains a list of absolute GPS coordinates $\Lambda_u(\lambda_u^1, \lambda_u^2, \ldots\ldots, \lambda_u^n)$, and the location of current social event is Λ_e, therefore, we build a new list consisting of the distance value as $Dis(\lambda_u^i, \Lambda_e), 1 < i < n$, where

$$Dis(\lambda_u^i, \Lambda_e) = |\lambda_u^i - \Lambda_e|. \tag{2}$$

Assuming there is a probability distribution function $\Gamma(*)$ that measures the data distribution of $Dis(\lambda_u^i, \Lambda_e)$, we then represent the individual mobility pattern (ϕ) and collaborative mobility pattern (Φ) as

$$\phi \sim \ \Gamma(Dis(\Lambda_u, \Lambda_e)) \quad u \in U_{all} \tag{3}$$

$$\Phi \sim \ \sum_{u \in \Omega} \Gamma(Dis(\Lambda_u, \Lambda_e)). \tag{4}$$

Finally, given the event's collaborative mobility pattern Φ and the user's individual mobility pattern ϕ, we apply KL-divergence [19] to calculate the difference between these two probability distributions. Thus, the mobility influence F_{MI} is defined as follows:

$$F_{MI} = \sum \phi \log \frac{\phi}{\Phi}. \tag{5}$$

where KL-divergence measures the difference between two probability distributions. A smaller value of F_{MI} indicates a higher similarity between Φ and ϕ, thus a higher possibility that the corresponding user could be on-site w.r.t the target social event.

4.2 Content Similarity

Each social event usually has a theme that can be described by the topic of the event. For example, a pop music concert held by Justin Bieber has its own topic that could contain words like "Justin Bieber" and "Pop Music", etc. And on-site users are more likely to talk/tweet about these words on social media than those who are not on-site. Therefore, we propose a text-based similarity measure among users' historical tweets and event topics to quantify the content similarity.

Without loss of generality, the event topic Θ can be modeled using the tweets that are generated from its participants Ω during the event time window, and we represent the event topic based on the keywords extracted from those tweets. Similarly, the keywords extracted from the event-related users' tweets that are posted before/during the event can be extracted to model the users' historical topic θ. And a higher similarity between these two topic features indicates a higher likelihood that the corresponding event-related user would attend the event or be present near the event location.

Text similarity has been a subject of extensive research. One widely used method is Latent Dirichlet Allocation (LDA) [20], which is a probabilistic topic model that can be employed in text similarity calculation. Another approach is to use an external lexical database, such as WordNet, to mine the relationships among words [21]. However, these methods ignore the temporal influence of different text. Intuitively, among the historical tweets posted by the user prior to the event, the more recent tweets should carry more weight. As such, we extract individual historical tweets topic using a time fading model that uses higher weights for more recent posts. Specifically, the function of time fading keywords extraction is shown as follows:

$$\theta = < kw_1, kw_2, \ldots\ldots, kw_n > \tag{6}$$

$$kw_i = \sum_{j=1}^{m} w_{i,j} \times \exp\{-(T_e - t_j)\}, \tag{7}$$

where θ contains n weighted keywords (kw) that describe the user's historical tweets topics, and kw_i is calculated using Eq. 7, in which $w_{i,j}$ represents the frequency of the ith keyword occurring at time t_j, and T_e is the start time of the target social event. Similarly, event topic Θ can be represented by a vector using the words that were tweeted by its participants. Therefore, the content similarity F_{CS} between each individual's historical tweets topic θ and the event topic Θ is defined as follows:

$$F_{CS} = \frac{\Theta\theta}{|\Theta||\theta|}, \tag{8}$$

where a higher value of F_{CS} indicates a higher probability that the user is attending the social event (i.e., on-site).

4.3 Social Relationship

As discussed in previous work [6], a user would review a product that is already reviewed by his/her friends. And [7] also demonstrates that friendship has a major impact on people's attendance in real-world events. In general, if many friends of a user are attending a target event, that user is more likely to participate in the event as well. In

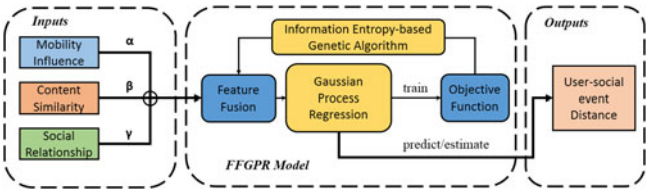


Fig. 3. The workflow of the FEGOR model.

this paper, we consider social relationship as a factor to estimate the *User-Social Event Distance*.

To model the social relationship of a user to the social event, we first obtain the list of known participants Ω based on active users' proximity to the event location since active users expose their locations. However, due to the limitation of TwitterAPI that we used in our work, we are prohibited from directly obtaining the users' friend list. To solve this problem, we employ user's mentions behavior on tweets to reflect the relationship to some extents, since previous work [53] has discussed that user behavior patterns can be used to build their social network. Hence, we could obtain the estimate friend ω of a user through mentions. After that, we have two sets of users: Ω contains the participants of the event and ω consists of a list of users recently mentioned by event-related users. Finally, the social relationship F_{SR} is defined as follows:

$$F_{SR} = \frac{|\Omega \cap \omega|}{|\Omega|}, \tag{9}$$

where $|\Omega \cap \omega|$ is the number of event-related users' friends who participated in the event, and the $|\Omega|$ represents the total number of participants. Hence, a higher F_{SR} indicates a higher probability that the user could be on-site, since many of the participants are his/her friends.

5 Inference Algorithm

We propose a Fused fEeature Gaussian prOcess Regression (FEGOR) model to estimate the *User-Social Event Distance*. And an information entropy based genetic algorithm is proposed for parameter learning. The workflow of the FEGOR model is illustrated in Fig. 3.

5.1 Gaussian Process Regression Model

Previous studies have shown that human mobility can be modeled by Brownian motion [39], [40], [41], and the jump length of Brownian motions exhibits a Gaussian distribution [42]. Based on this observation, we model the distance between user's trajectory and social event location as a Gaussian distribution, whose mean and variance can be influenced by miscellaneous factors. In this work, we aim to determine some influential factors by extracting user's mobility, content and social relationship features. Hence, we apply Gaussian process regression (GPR) [22] as the inference model for our work since every user's mobility pattern obey Gaussian distribution, and their combination has a multivariate normal distribution. Based on GPR model, we are able to infer an unknown target output y_* conditioned on the known value of y, x; and its corresponding input x_*, where y_* is the distance that need to be estimated, x_* is the integration of three features mentioned above, y and the corresponding x are the existing knowledge which is leaned from the *Active Users*. In other word, we can infer the parameters for Gaussian distribution of one unknown user using the parameters from known users.

In general, consider a finite set of input $X = \{x_1, x_2, \ldots, x_m\}$, we assume that for any such set there is a covariance matrix K with elements $K_{i,j} = k(x_i, x_j)$, in which we apply squared exponential covariance function in our model. And this function specifies the covariance between pairs of random variables:

$$k(x_i, x_j) = \exp\Big(-\frac{1}{2}|x_i - x_j|^2\Big), \tag{10}$$

and for each input x, there is a corresponding output y that is generated by

$$y = f(x) + \epsilon, \tag{11}$$

which is the noisy version in modeling observations based on function values, where $f(x)$ and ϵ are both normal random variables, with prior distribution that $f(x) \sim N(0, K(X, X))$ and $\epsilon \sim N(0, \delta_m^2)$. Thus, the joint distribution of the training output y and the test output y_* according to the prior is

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\Bigg(0, \begin{bmatrix} K(X,X) + \delta_m^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\Bigg). \tag{12}$$

After that, it is easy to generate the distribution function of y_* in probabilistic terms, corresponding to conditioning the joint Gaussian prior distribution on the observations to give:

$$y_* | X, y, X_* \sim \ N(\widetilde{y_*}, cov(y_*)), \tag{13}$$

where $\widetilde{y_*}$ and $cov(y_*)$ are derived as follows:

$$\begin{aligned} \widetilde{y_*} &= E[y_* | X, y, X_*] \\ &= K(X_*, X)[K(X,X) + \delta_m^2 I]^{-1} \end{aligned} \tag{14}$$

$$\begin{aligned} cov(y_*) = {} & K(X_*, X_*) - \\ & K(X_*, X)[K(X,X) + \delta_m^2 I]^{-1} K(X, X_*). \end{aligned} \tag{15}$$

Finally, the parameters for distribution of y_* are obtained and we employ its mean value $\widetilde{y_*}$ as the estimated value according to the input x_*. And it has been proved that the mean value for Gaussian distribution can minimize the risk for the loss function [22], where the objective function to optimize in our system is

$$\arg\min_{y_{predict}} \sum (y_{predict} - y_{real})^2. \tag{16}$$

In our model, $\widetilde{y_*}$ is for $y_{predict}$ and y_{real} is the ground truth w.r.t $y_{predict}$. Based on our modeling process above, to estimate a target output y_*, the key is determining the distance/similarity between X_* and X, and elements in the covariance matrix K should have high distinguish power for all possible situations. However, as pointed out by Beyer et al. [54], distance functions would lose their usefulness in high dimensions. In other words, compared with lower dimensional spaces, the distance/similarity among different samples becomes indiscernible in high dimensional spaces, which could cause the model to fail. To avoid such situation, our model transforms the input x into a

TABLE 1
Description of Tweet Raw-Data

Property	Description
uid	distinct user id for each user
tid	distinct tweet id for each tweet
content	tweet content that was posted by the corresponding user
mentions	Each tweet may mention some of the user's friends, thus this information contains the friends' ids that the user mentioned in his/her recent tweets
coordinates	Each tweet may have a specific location, if the user exposes his/her location. Each coordinate consists of longitude and latitude e.g., (-73.99880076, 40.75140471) or it would be null
creation time	The corresponding time when the tweet was posted, e.g., 2014-11-25 10:57:42

lower-dimensional space by applying linear combination of the proposed features as follows:

$$x = \alpha * F_{MI} + \beta * F_{CS} + \gamma * F_{SR}, \tag{17}$$

where F_{MI}, F_{CS} and F_{SR} represent the Mobility Influence, Content Similarity, and Social Relationship respectively with parameters α, β and γ indicating the weights of these features as shown in Fig. 3. Intuitively, these three features may have different influences for different social events. It is thus important that we tune the parameter values for different events to achieve more accurate user-social event distance estimation, and the tuning method is illustrated next.

5.2 Parameter Learning

We apply Genetic Algorithm (GA) [23] in our work to learn the most suitable parameters. Instead of randomly initializing the first generation of population for searching, we utilize the information entropy [24] of features to build "seed" for possible solutions. Specifically, the "seed" for setting up α, β and γ are generated as follows:

$$H(x) = -\int p(x_i)\log p(x_i)dx, \tag{18}$$

where $p(x)$ represents the probability density function for each of the three features. The information entropy based genetic algorithm for parameter learning is illustrated in Algorithm 1. In details, the algorithm mainly consists of four typical procedures: *Initialization*, *Selection*, *Crossover*, and *Mutation*. Different from traditional method for population initialization, we first compute the information entropy of each feature as "seed", which is a vector containing values of α, β, and γ, to initialize the first generation of population. We then obtain the fitness value using objective function in Equation 16, and proceed to apply selection, crossover, and mutation on the population. Besides, to avoid useless searches, we add a function named "*RemoveDuplication*" to delete the duplicate individuals. And the algorithm terminates after a fixed number of iterations. Specifically, we apply linear ranking [34], [35] selection scheme, single-point crossover [36] method, and flip bit [37] technique in our selection, crossover and mutation function respectively.

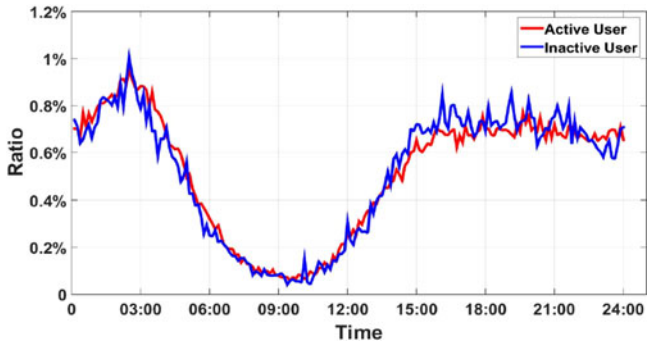


Fig. 4. The distribution of tweet's creation time.

Algorithm 1. Information Entropy based Genetic Algorithm

Input: *FeatureSet*, N_{Pop}, N_{Iter}, $Range$, $Rate_c$, $Rate_m$
Output: α, β, γ
1: *Seed* $\Leftarrow$ InformationEntropy(*FeatureSet*)
2: $Poplist \Leftarrow$ Initialization(*Seed*, N_{Pop})
3: **For** i **In** N_{Iter} **Do**
4: ObtainFitnessValue($Poplist$)
5: $threshold \Leftarrow$ Math.Random($Range$)
6: $Poplist_s \Leftarrow$ Selection($Poplist$, $threshold$)
7: $Poplist_c \Leftarrow$ CrossOver($Poplist_s$, $Rate_c$)
8: $Poplist_r \Leftarrow$ RemoveDuplication($Poplist_s$)
9: $Poplist_m \Leftarrow$ Mutation($Poplist_r$, $Rate_m$)
10: $Poplist \Leftarrow Poplist_m$
11: **End**
12: **Return** α, β, γ from BestSolution($Poplist$)

6 Experimental Results

6.1 Dataset Description

Using the TwitterAPI, we collected raw data of user tweets that were posted at the New York City during the period of Nov 1, 2014 and Jan 13, 2015. Each tweet contains creation time, content, mentions, coordinates and the corresponding user id. In total, we have collected 6,727 users and 1,187,847 tweets. The details of the dataset are shown in Table 1.

To check whether there are significant bias between active and inactive users, we also compare the behavioral patterns in terms of the tweet's creation time distribution and some other statistical metrics as shown in Fig. 4 and Table 2 respectively. The distribution of tweet's creation time shows a similar pattern between active and inactive users. According to the measurements in Table 2, we can observe that the only significant bias is the ratio of geo-tagged tweet, and it is natural to have such difference since active users are defined to be more active in posting geo-tagged tweets than others. Apart from this, the behavioral patterns measured by other metrics

TABLE 2
Comparison of Behavioral Patterns

User Set	#Words per Tweet	#Mentions per Tweet	Ratio of Geo-tagged Tweets
Active User	12.5876	0.7625	**0.3485**
Inactive User	13.1423	0.8703	**0.0437**

TABLE 3
Summary of Identified Social Events

Time	Topic Word	Location(NYC)
25 Nov 03am	ferguson	Union Square
25 Nov 10am	ferguson	Union Square
26 Nov 09am	ferguson	Williamsburg Bridge
26 Nov 10am	ferguson	Stuyvesant Square
26 Nov 12am	ferguson	Time Square
29 Nov 08am	ferguson	Columbus Ave&106th St
04 Dec 10am	ferguson	Time Square
05 Dec 08am	ferguson	City Hall Park
05 Dec 10am	ferguson	Tweed Courthouse
14 Dec 03am	blacklivesmatter	Washington Square Park
14 Dec 04am	blacklivesmatter	Washington Square Arch
14 Dec 06am	blacklivesmatter	Union Square
14 Dec 08am	blacklivesmatter	Tweed Courthouse
14 Dec 10am	blacklivesmatter	Tweed Courthouse

are almost the same, which indicates that those two types of users follow similar behavioral patterns.

Based on the collected data, we construct the feature model and implement our distance estimation method. Specifically, we first identify social events using topic words and the locations of those events using the method proposed in [11], which results in 14 social events during our data collection period.

Table 3 summarizes the 14 social events that we have identified. Since we use one hour as the moving time window to detect topic words, some prolonged social events may be separated by multiple time windows. As shown in Table 3, one prolonged protest parade on Dec 14 started at 3am and ended at 10am, and the protesters moved from Washington Square Park to Tweed Courthouse.

6.2 Detailed Results

6.2.1 Locating the Social Event

We adopt the algorithm in [11] for event localization. The method originally considers "local words" to match the locations with content, which can be adapted for our system by replacing the "local words" with "topic words". Topic words refer to the hot/key words that are generated by users near a social event.

For each topic word, given a potential center, the central frequency c and dispersion parameter η, we determine the event location as follows. For each observed user's location, suppose all users tweeted the topic word from that location a total of n times, then we multiply the overall probability by $(cd_i^{-\eta})^n$, where $d_i^{-\eta}$ is the distance between the user location and the center of the topic word; if no users in that location tweeted the topic word, we multiply the overall probability by $1 - cd_i^{-\eta}$. To reduce the computation complexity, we use logarithms of probabilities instead of multiplying probabilities. For example, let Δ be the set of occurrences for the topic word, let d_i be the distance between a location and the center of the topic word, then we need to maximize the likelihood function as follows to determine the potential center of the topic word, as well as the location of the target event:

$$f(c,\eta) = \sum_{i \in \Delta} \log cd_i^{-\eta} + \sum_{i \notin \Delta} (1 - cd_i^{-\eta}). \tag{19}$$

TABLE 4
Detail Information of Three Selected Social Events

Event ID	Description
1	25 Nov 10-11am, 2014. A group of people gathered together at Union Square and then marched on 5th Ave to protest the death of Michael Brown, a black person who was killed by law enforcement officers in Ferguson, Missouri [26].
2	26 Nov 12-13am, 2014. Hundreds of people marched in Manhattan from Union Square to Time Square. They peacefully protested a grand jury's decision not to indict a white police officer who fatally shot an unarmed black teenager in Missouri [31].
3	14 Dec 03-04am, 2014. A large amount of people marched through Manhattan in the largest protest in New York City since a grand jury declined to indict an officer in the death of an unarmed black man on Staten Island. They gathered at Washington Square and the crowd thunderous chants of "Hands up! Don't shoot!" and "Justice Now!" [32], [33].

Backstrom et al. [25] have proved that $f(c,\eta)$ has exactly one local maximum over its parameter space, that is if a center is chosen, we can iterate c and η to find the largest $f(c,\eta)$ value, thus we can obtain the corresponding location of the target event using its topic words. Since the algorithm requires sufficient data to discover the location of social events, small events tweeted by a few people are ignored. As a result, we have detected 14 relatively large/influential events in our dataset, and the average on-site user amount is more than 140.

We consider three distinct social events illustrated in Table 4 as examples. And the event center locating performance of event 1 is shown in Fig. 5, in which active users represent the users who posted or re-posted tweets about "ferguson", while non-active users are users who did not post anything about "ferguson" during the time period. We can see that, a large number of people who tweeted about "ferguson" were in Manhattan, New York City. However, the real-world event only took place near Union Square.

Leveraging the event localization algorithm, we can locate the event center precisely, in which the estimated location is (-73.99052107, 40.73649499), the center frequency c is 0.99 and the dispersion factor η is 0.437.

As shown in Table 5, the three distinct social events have separate locations, central frequencies, and dispersions as determined by the algorithm. The central frequency for the three events are all 0.99 which means that the possibility of a user in the event location tweeting about the event is relatively high but they have different dispersion parameters,

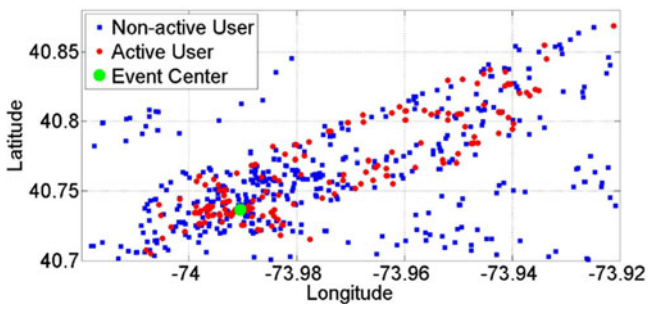


Fig. 5. The event localization result for event 1.

TABLE 5
Event Localization Results of Three Different Events

Event ID	Time	Location	c	η
1	25 Nov 10-11am	-73.991,40.736	0.99	0.44
2	26 Nov 12-13am	-73.986,40.758	0.99	0.74
3	14 Dec 03-04am	-73.997,40.731	0.99	0.52

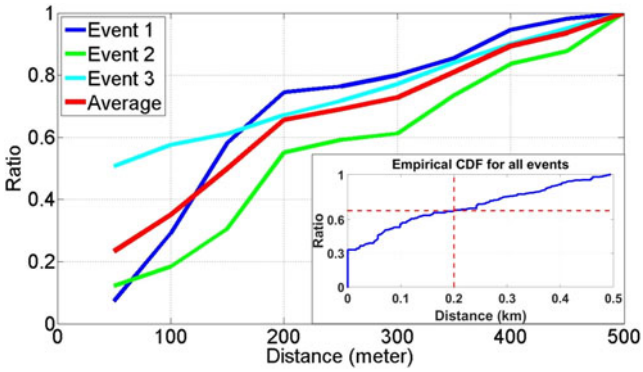


Fig. 6. The choice of radius for events.

indicating that different events have different influence range, and a higher value for η means the corresponding event has a smaller radius.

6.2.2 Feature Evaluation

Determining Participants and Historical Records. Given the location of each event, the corresponding participants are defined to be persons who stay within a certain radius from the event location. Without loss of generality, the radius for a social event is usually lesser than 500 meters. However, different event may differ in the area range it covers. Therefore, we need to select an appropriate range of radius to cover most of the situations to model the events' area. Fig. 6 shows the ratio of event-related users covered within certain radius of the event location. The figure not only shows statistic results of the three events demonstrated in Table 4 separately, but also illustrates the statistics for all 14 events in Table 3 in its inner figure.

It can be observed that a "knee point" appears at around 200-meter radius, which covers more than half of the event-related users and average ratio of user coverage is above 0.6. Based on this observation, we set the radius of 200 meters to determine the known participants for each social event. In other words, given a social event, its known participants are those whose user-social event distances are lesser than 200 meters. As for each user, the number of historical tweets that we are able to crawl is limited to 200 according to Twitter Developer Documentation [44] per distinct request. Therefore, we use every user's 200 historical tweets as default to model user's individual patterns.

Mobility Influence. Fig. 7 shows the correlation between user-social event distance and the mobility influence, which is represented by KL-divergence as described in Section 4.1. More specifically, the KL-divergence measures the similarity between event's collaborative mobility pattern Φ and individual mobility pattern ϕ, in which a smaller value indicates a higher similarity between these two features thus can be utilized to infer a user is more likely to be on-site than those who have high KL-divergence values. As can be seen in the figure, a larger user-social event distance corresponds to a larger KL-divergence value, which proves that KL-divergence can be an effective factor describing the distance between users and an event.

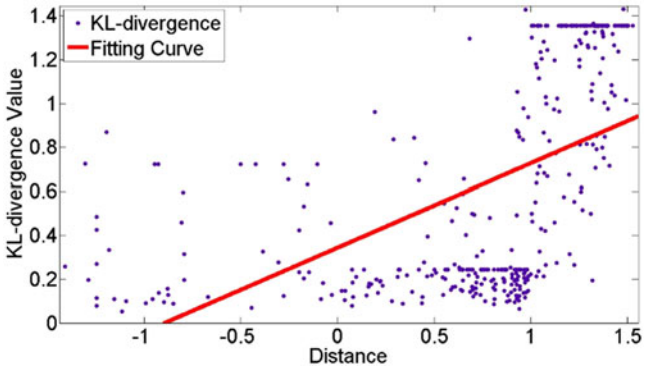


Fig. 7. KL-divergence versus user-social event distance.

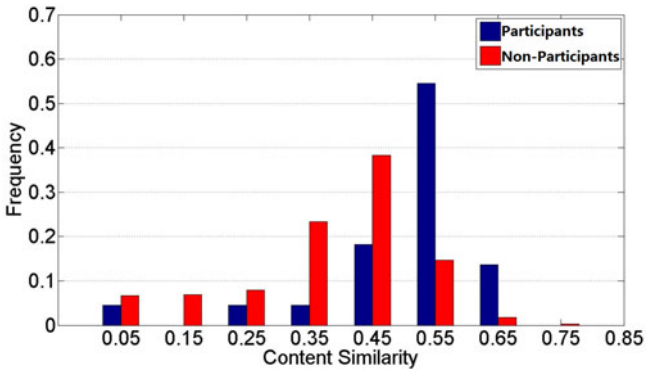


Fig. 8. The Content Similarity with different users.

Content Similarity. We calculate the content similarity for both participants and non-participants w.r.t social event, after that, we bin the results into 9 equally spaced containers (starting from 0.05 with 0.1 step size), and count the number of results in each container. As shown in the histogram in Fig. 8, over 50 percent of the participants' content similarity is around 0.55 while only 10 percent of the non-participants (a subset of event-related users whose locations are far away from the event location) have the same content similarity value. Most (83.29 percent) non-participants' content similarity is less than 0.55 while 68.18 percent of the participants have larger than 0.55 content similarity values. This difference between participants and non-participants proves that content similarity also contributes to user-social event distance estimation.

Social Relationship. Fig. 9 shows the difference between participants and non-participants in terms of social relationship. Based on the figure, over 70 percent of the participants' social relationship score is larger than 0.5 while less than 20 percent of the non-participants have larger than 0.5 social relationship scores. Thus, if a user's friends are participating in a target event, we can infer with high confidence that the user is likely to be present near the event.

6.2.3 Inferring the Parameters for Genetic Algorithm

After we obtain the three features calculated from event patterns and event-related users' patterns, we then infer the model parameters based on the training dataset. As

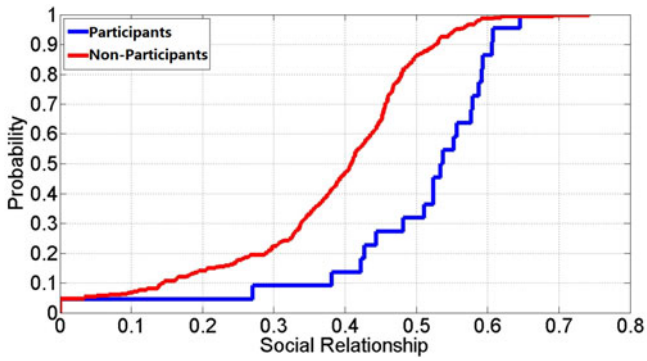


Fig. 9. Social Relationship between different users.

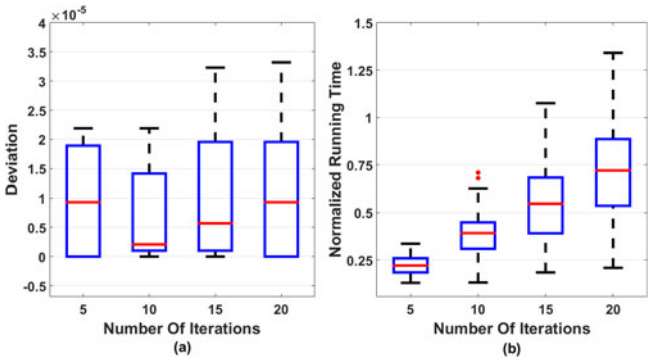


Fig. 10. Evaluation on number of iterations.

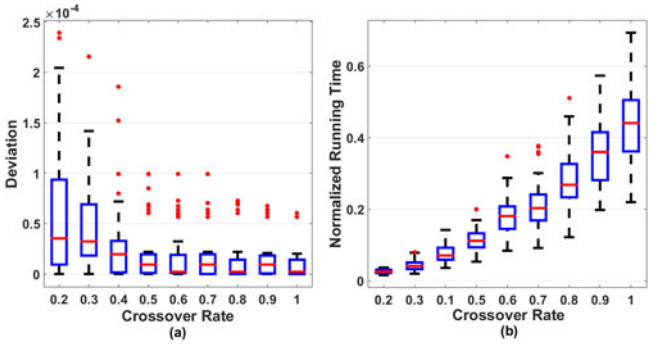


Fig. 11. Evaluation on crossover rate.

mentioned above, the parameters α, β and γ in Equation 17 determine the different influence weights for constructing the input of the FEGOR model. Specifically, people may be attracted to different social events due to assorted reasons. For example, an invitation from friends (social relationship) or public advertisements (content preference) of a specific social event could be influential factors that lead to a person's attendance at the social event. And the mobility pattern of a person could also affect the possibility he/she participates in some social events. Therefore, it is meaningful to figure out the different influences of the proposed three features with respect to distinct kinds of events.

Before we start to learn the weights across the three features, there are a few parameters that we need to determine in our information entropy based genetic algorithm as illustrated in Algorithm 1. In order to evaluate the overall performance of the genetic algorithm, we first apply exhaustive search (known as brute-force search [43]) method to traverse all possible combinations of parameter α, β, and γ to

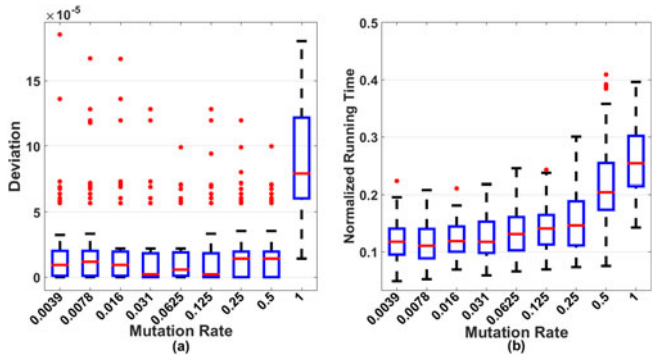


Fig. 12. Evaluation on mutation rate.

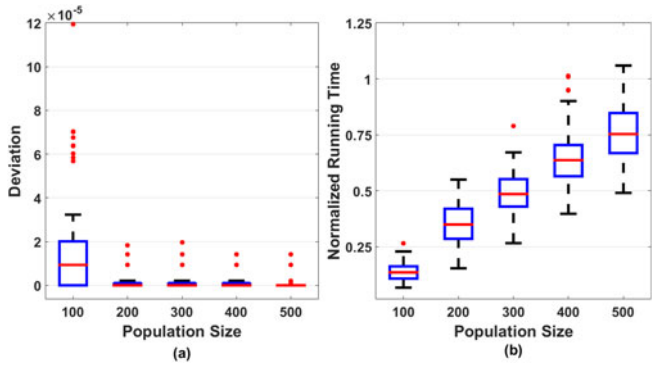


Fig. 13. Evaluation on initial population size.

get the optimal solution. In which we obtain the Mean Absolute Error (MAE) using the optimal solution for regression on our test data set, and we also record the running time when applying exhaustive algorithm in searching the best solution. We then compute the deviation between the MAE obtained by our genetic algorithm and exhaustive algorithm to show how well our genetic algorithm could perform. The smaller the deviation is, the better the genetic algorithm performs. In addition, we compare the running time between these two algorithms in a normalized manner. Specifically, we consecutively evaluate four main parameters: *number of iterations*, *crossover rate*, *mutation rate*, and *initial population*. We evaluate each one of them individually while setting others as static value according to [38].

The detailed evaluation results are illustrated in Figs. 10, 11, 12, and 13. From those figures, we can observe that with the increasing of number of iterations, crossover/mutation rate, and initial population size, the running time of genetic algorithm increases as well. As for deviations, changes on different parameter has different patterns. In specific, there is no significant difference among different number of iterations, and that is almost the same when considering changes on mutation rate lower than 0.5. However, crossover rate reduces the deviation as its value increase when lower than 0.5, and it appears no further improvement when its value is larger than 0.5. And the influence on population size shows similar pattern, it remains stable when larger than 200. In summary, the deviation across all tuning processes are relatively small. Considering the complexity of our work in real world tasks, we prefer a time-efficient algorithm with low errors. Therefore, in our evaluations, we finally set the number of iterations, crossover rate, mutation rate, and initial population size as 5, 0.5, 0.01, and 200, respectively.

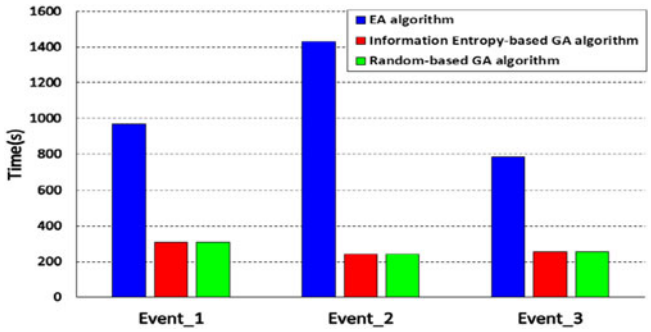


Fig. 14. Case study: running time across different algorithms.

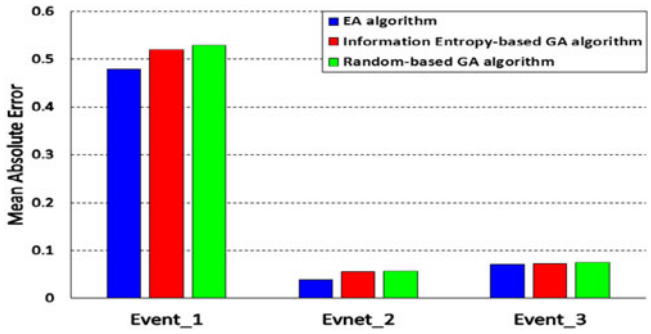


Fig. 15. Case study: MAE across different algorithms.

TABLE 6
Results of Parameter Learning for Different Events

Event			Parameters		
ID	Location	Time	α	β	γ
1	Union Park & 5th Ave	25 Nov 10-11am	0.33	**0.42**	0.25
2	7th Ave	26 Nov 12-13pm	**0.39**	**0.41**	0.20
3	Washington Square	14 Dec 03-04am	**0.40**	0.21	**0.39**

We further provide case studies in Figs. 14 and 15 on the three events that are described in Table 4. We implement the algorithms using the determined parameters on a workstation with 8 GB memory and Intel i5 processor. Since we terminate the genetic algorithm with fixed number of iterations, the time costs for information entropy based GA is same with random based GA. While exhaustive algorithm has much longer running time, and in specific, genetic algorithms can usually finish the task within 6 minutes. And in Fig. 15, we observe that our information entropy based GA can achieve lesser MAE than random based GA, which indicates the "seeds" generated by feature information entropy could lead the genetic algorithm to areas where optimal solution are likely to be found than random based "seeds".

6.2.4 Feature Weight Analysis and Distance Estimation

Analyzing Feature Weight. In order to better understand the reasons/factors that may motivate ordinary people to participate in social events, we illustrate the learned feature weight, values for parameter α, β, and γ, of the three events mentioned above in Table 6. Different events have different parameter values indicating that people attracted to the events are influenced by the three proposed factors differently. As for event 1, we can see that β is the largest among the three factors which indicates that people attend that event mainly because of the content preference. And for event 2, apart from content similarity, α also has a relatively high value indicating the importance of user mobility pattern to motivate users' attendance of the target event. In specific, we find that the participants of event 2 marched from Union Square to Time Square as described in Table 4, which also indicates that they could have similar mobility patterns. And for event 3, we can see that social relationship (γ) also plays a significant role, knowing that mobility influence (α) earns the largest but with almost equally value: 0.40 versus 0.39. Since event 3 happened at 3am, it is reasonable to believe that users' participation in this event was influenced more strongly by their friends who were attending the same event.

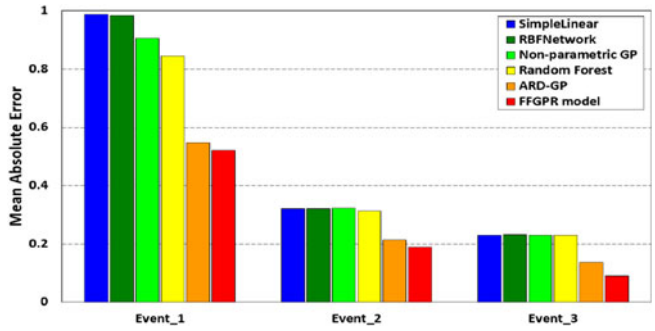


Fig. 16. MAE of user-social event distance estimation using different models.

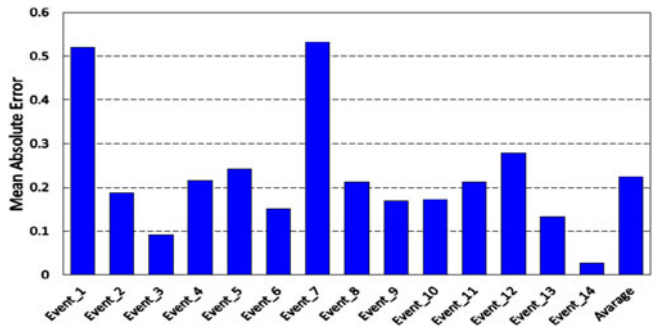


Fig. 17. MAE of user-social event distance estimation for all 14 events.

Distance Estimation and Model Comparison. Using the FEGOR model and the parameter values learned for a target social event, we can then predict/estimate the user-social event distance for every event-related individuals. Fig. 16 compares the distance estimation performance (measured by MAE) of our FEGOR model and several other regression models, including SimpleLinear Regression [46], RBFNetwork [45], Random Forest [51] and traditional Gaussian Regression [22]. Specifically, multi-variate Gaussian Regression model has been studied in many works [48], [49], [50], which can be a competitive model to ours. Hence, we implement such model using the ARD kernel function through the GPy toolkit [52], and also compare our model with it as shown in Fig. 16.

It can be observed that our FEGOR model outperforms the other models for three different events. Specifically, ARD-GP model (multi-variate GP) achieves closely results comparing to ours, however, FEGOR still improves nearly 5, 12, and 34 percent on event_1, event_2 and event_3 respectively (17 percent average improvement).

TABLE 7
Correlation between Different Factor and MAE

Factor	PCC	p-value
Entropy of mobility influence (E_{MI})	**0.8025**	**0.0006**
Entropy of content similarity (E_{CS})	0.0677	0.8182
Entropy of social relationship (E_{SR})	**0.8276**	**0.0003**
Number of user in testing dataset (N_U)	**0.8309**	**0.0002**

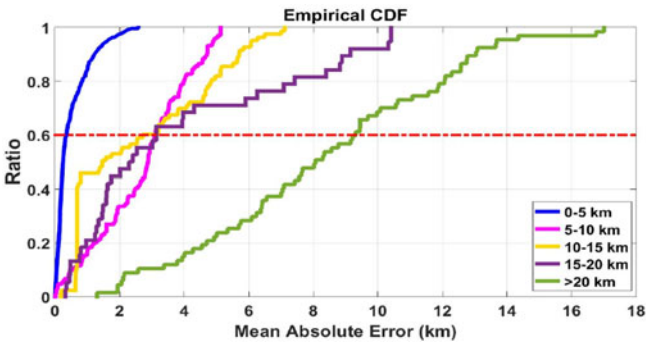


Fig. 18. Distributions of MAE for different real User-Social Event Distance ranges.

Furthermore, we evaluate our model for all 14 social events that are discovered in our dataset, and the prediction performance for each event is shown in Fig. 17. We can see that 12 out of 14 (85.7 percent) events have MAE values under 0.3 excepting Event_1 (0.5214) and Event_7 (0.5325). However, the performance across all the 14 events fluctuates according to Fig. 17.

To investigate the reasons that may causes such fluctuation, we consider the entropy of the three proposed features and the amount of users in testing dataset of each social event. Specifically, for each social event, the entropy of one feature represents the diversity of such feature value distribution. And we apply Pearson correlation coefficient (PCC) [47] to measure the relationship between these factors and the MAE values of all 14 events. The correlation results are shown in Table 7, and it can be observed that most of the factors have significant impact on MAE values. In particular, E_{MI}, E_{SR}, and N_U all have positive correlation with MAE (PCC higher than 0.8), which indicates that increasing the entropy of mobility influence, social relationship, and the number of user in testing dataset will also increase the prediction risk. In general, higher entropy means higher uncertainty and the data will become more unpredictable.

However, despite how much uncertainty the data sets have, our proposed method can still provide us with reasonable estimation results. To further illustrate the performance of our model under varying distance, Fig. 18 shows the distributions of MAE for different real *User-Social Event Distance* ranges, and the red dotted line indicates 60 percent. We can observe that for users whose real *User-Social Event Distance* is between 0 and 5 km, 60 percent of the MAEs are lower than 0.3 km. And for users whose real *User-Social Event Distance* is in the range of 5-10, 10-15, 15-20, and >20 km, the 60 percent MAE value is lower than 3.03, 2.54, 3.02 and 9.22 km, respectively. As shown in Fig. 18, our model is much more effective in distance estimation when a user is close to an event (0-5km) than when a user is far from an event. Since on-site users are usually not far from an event, our model is sufficient to be implemented for real-world applications. And the average MAE for all events and users is around 1.7 km ($10^{0.227}$) according to Fig. 17.

TABLE 8
Comparison Summary

Method	Features			MAE(km)
	Mobility	Content	Social	
FEGOR	**Yes**	**Yes**	**Yes**	**1.7**
Cheng [11]	No	Yes	No	200
OLIM [13]	Yes	Yes	No	28.5
Backstrom [25]	Yes	No	Yes	41.9
Ryoo [12]	Yes	No	Yes	10

Finally, we compare the proposed method with a few existing methods listed in Table 8 in terms of the features that are used in the model and the MAE of the estimation/prediction results. Compared with other methods that use partial features, our proposed model integrates three distinctive features and achieves much better performance. According to all the experimental results above, our model is more effective than others approaches. The main reasons for our model's improved performance are the following. First, Gaussian Process is inherently suitable for modeling human mobility trajectories as discussed in [39], [40], [41]. Second, we integrate all the extracted features rather than applying them partially. Third, we apply linear combination among the features to make it more distinguishable in a lower-dimensional space when determining the similarity between different samples. And finally, we utilize an information entropy based parameter learning algorithm to deduce the most effective combination of parameters, which leads to the success of our FEGOR model.

7 Conclusion

In this paper, we have proposed a Fused fEature Gaussian prOcess Regression (FEGOR) model, which effectively incorporates three features: mobility influence, content similarity and social relationship that are extracted from social events with event-related users, to address the problem of estimating the user-social event distance when users do not disclose their location profiles. At the same time, since we transform all absolute location data into distance values, we not only achieve our goal in this paper, but also protect users' location privacy in a coarse-grained level.

Different from those existing works that only attempt to detect social events using social networks, our work offers new insights into the motivating factors of users attending social events that contribute to estimating the user-social event distance.

This work builds upon previous location based research and focuses on finding on-site users w.r.t social events, which can be of immense value for many social-oriented applications. For example, the proposed framework can be leveraged by government officials to better understand the patterns of different social events (e.g., parade, protest) and devise better crowd management strategies. And based on the testimony from discovered on-site users, governments or police forces can obtain more detailed and meaningful

information of the target event. As for our future work, we plan to expand our model beyond Twitter and consider other types of social network platforms.

Acknowledgments

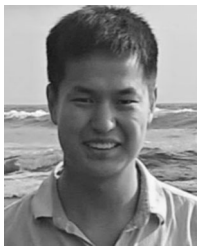
This work was supported in part by the National Science Fund for Distinguished Young Scholars (No. 61725205), the National Basic Research Program of China (No. 2015CB352400), the National Natural Science Foundation of China (No. 61332005, 61772428), and the US National Science Foundation (No. 1528138).

References

[1] [Online]. Available: http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/
[2] B. Hecht, L. Hong, B. Suhand, and E. H. Chi, "Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2011, pp. 237–246.
[3] D. Jurgens, "That's what friends are for: Inferring location in online social media platforms based on social relationships," in *Proc. Int. Conf. Weblogs Social Media*, 2013, pp. 273–282.
[4] D. Yang, D. Zhang, B. Qu, and P. Cudrè-Mauroux, "PrivCheck: Privacy-preserving check-in data publishing for personalized location based services," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 545–556.
[5] R. Du, Z. Yu, T. Mei, Z. Wang, Z. Wan, and B. Guo, "Predicting activity attendance in event-based social networks: Content, context and social influence," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 425–434.
[6] Y. Rong, H. Cheng, and Z. Mo, "Why it happened: Identifying and modeling the reasons of the happening of social events," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1015–1024.
[7] P. I. Georgiev, A. Noulas, and C. Mascolo, "The call of the Crowd: Event participation in location-based social services," in *Proc. 8th Int. AAAI Conf. Weblogs Social Media*, 2014, pp. 141–150.
[8] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Who, where, when and what: Discover spatio-temporal topics for Twitter users," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 605–613.
[9] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 68–76.
[10] N. Dalvi, R. Kumar, and B. Pang, "Object matching in tweets with spatial models," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 43–52.
[11] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating twitter users," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag.*, 2010, pp. 759–768.
[12] K. M. Ryoo and S. Moon, "Inferring twitter user locations with 10 km accuracy," in *Proc. Companion Publication 23rd Int. Conf. World Wide Web Companion Int. World Wide Web Conf. Steering Committee*, 2014, pp. 643–648.
[13] Y. Yamaguchi, T. Amagasa, H. Kitagawa, and Y. Ikawa, "Online user location inference exploiting spatiotemporal correlations in social streams," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag.*, 2014, pp. 1139–1148.
[14] A. Sadilek, H. Kautz, and J. P. Bigham, "Finding your friends and following them to where you are," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 723–732.
[15] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2013, pp. 344–353.
[16] Y. Wang and M. S. Kankanhalli, "Tweeting cameras for event detection," in *Proc. 24th Int. Conf. World Wide Web. Int. World Wide Web Conf. Steering Committee*, 2015, pp. 1231–1241.
[17] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 851–860.
[18] A. Tarasov, F. Kling, and A. Pozdnoukhov, "Prediction of user location using the radiation model and social check-ins," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, Art. no. 8.
[19] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[21] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.
[22] C. E. Rasmussen and C. K. I. Williams, in *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006
[23] M. Mitchell, in *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
[24] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
[25] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 61–70.
[26] [Online]. Available: http://edition.cnn.com/2014/11/25/us/national-ferguson-protests/
[27] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.
[28] Z. Yu, H. Wang, B. Guo, T. Gu, and T. Mei, "Supporting serendipitous social interaction using human mobility prediction," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 6, pp. 811–818, Dec. 2015.
[29] C. Chen, D. Zhang, N. Li, and Z.-H. Zhou, "B-Planner: Planning bidirectional night bus routes using large-scale taxi GPS traces," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1451–1465, Aug. 2014.
[30] Daqing Zhang, Bin Guo, and Zhiwen Yu, "The emergence of social and community intelligence," *Comput.*, vol. 44, no. 7, pp. 21–28, Jul. 2011.
[31] [Online]. Available: http://www.nydailynews.com/news/politics/nyc-council-members-walk-protest-ferguson-grand-jury-article-1.2023804, Accessed on: Jun 5, 2017.
[32] [Online]. Available: https://www.nytimes.com/2014/12/14/nyregion/in-new-york-thousands-march-in-continuing-protests-over-garner-case.html, Accessed on: Jun 5, 2017.
[33] [Online]. Available: http://www.huffingtonpost.com/2014/12/13/millions-march-nyc_n_6320348.html, Accessed on: Jun 5, 2017.
[34] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proc. Int. Conf. Genetic Algorithms Their Applications*, 1985, pp. 101–111.
[35] J. J. Grefenstette and J. E, Baker, "How genetic algorithms work: A critical look at implicit parallelism," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 20–27.
[36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Boston, MA, USA: Addison-Wesley, 1989.
[37] S. N. Sivanandam and S. N Deepa, *Introduction to Genetic Algorithms*. Berlin, Germany: Springer Science, 2007.
[38] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst Man Cybernetics*, vol. TSMC–16, no. 1, pp. 122–128, Jan. 1986.
[39] T. Camp, J.f Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, 483–502, 2002.
[40] R. Groenevelt, E. Altman, and P. Nain, "Relaying in mobile ad hoc networks: The brownian motion mobility model," *Wireless Netw.*, vol 12, no. 5, pp. 561–571, 2006.
[41] S. Ioannidis and P. Marbach, "A brownian motion model for last encounter routing," in *Proc. Int. Conf. IEEE INFOCOM*, 2006, pp. 1–12.
[42] B. Jiang, J. Yin, and S. Zhao, "Characterizing the human mobility pattern in a large street network," *Phys. Rev. E*, vol 80, no. 2, 2009, Art. no. 021136.
[43] [Online]. Available: https://en.wikipedia.org/wiki/Brute-force_search
[44] [Online]. Available: https://dev.twitter.com/rest/reference/get/statuses
[45] D. S. Broomhead and D. Lowe, *Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks*, Malvern United Kingdom: Roy. Signals and Radar Establishment, 1988.

[46] A. Mehra, Statistical sampling and regression: Simple linear regression. PreMBA analytical methods. Columbia Business School and Columbia University, 2003, http://ci.columbia.edu/ci/premba_test/c0331/s7/s7_6.html
[47] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[48] M. K. Titsias and N. D. Lawrence,"Bayesian gaussian process latent variable model," in *Proc. 13th Int. Conf. Artif. Intell. Statistics*, 2010, vol. 9, pp. 844–851.
[49] T. Wang, "Adaptation of gaussian ARD kernel for multiclass classification," in *Proc. 8th Int. Conf. Fuzzy Syst. Knowl. Discovery.*, 2011, vol. 2, pp. 983–986.
[50] E. Snelson, Tutorial: Gaussian process models for machine learning. Gatsby Computational Neuroscience Unit, UCL, 2006, http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf
[51] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol 2, no. 3, pp. 18–22, 2002.
[52] GPy: A Gaussian process framework in python, [Online]. Available: http://github.com/SheffieldML/GPy
[53] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci. USA*, vol 106, no. 36, pp. 15274–15278, 2009.
[54] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *Proc. Int. Conf. Database Theory*, 1999, pp. 217–235.
[55] L. Wang, Z. Yu, B. Guo, T. Ku and F. Yi, "Moving destination prediction using sparse dataset: A mobility gradient descent approach," *ACM Trans. Knowl. Discovery Data*, vol 11. no. 3, 2017, Art. no. 37.
[56] F. Yi, Z. Yu, H. Chen, H. Du, and B. Guo, "Cyber-physical-social collaborative sensing: From single space to cross-space," *Frontiers Comput. Sci.*, Springer, Jan. 2018, https://link.springer.com/article/10.1007/s11704-017-6612-9
[57] B. Guo, H. Chen, Q. Han, Z. Yu, D. Zhang, and Y. Wang, "Worker-contributed data utility measurement for visual crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol 16, no. 8, pp. 2379–2391, Aug. 2017.
[58] B. Guo, Y. Liu, W. Wu, Z. Yu, and Qi Han, "Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Human-Mach. Syst.*, vol 47, no. 3, pp. 392–403, Jun. 2017.
[59] H. Chen, B. Guo, Z. Yu, L. Chen, and X. Ma, "A generic framework for constraint-driven data selection in mobile crowd photographing," *IEEE Internet Things J.*, vol 4, no. 1, pp. 284–296, Feb. 2017.

Zhiwen Yu received the PhD degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2006. He is currently a professor and the vice-dean of the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. He was a Alexander Von Humboldt fellow with Mannheim University, Germany, and a research fellow with Kyoto University, Kyoto, Japan. His research interests include ubiquitous computing and social network analysis. He is a senior member of the IEEE



Fei Yi received the BE degree in computer science and technology from Northwestern Polytechnical University, Xi'an, P.R. China, in 2010. He is currently working toward the PhD degree in computer science from Northwestern Polytechnical University, Xi'an, P. R. China. His research interests include ubiquitous computing, social network analysis, and data mining.

Qin Lv received the PhD degree in computer science from Princeton University, in 2006. She is an associate professor in the Department of Computer Science, University of Colorado Boulder. She is an associate editor of *ACM IMWUT*, and has served on the technical program committee and organizing committee of many conferences. Her main research interests are data-driven scientific discovery and ubiquitous computing.

Bin Guo received the PhD degree in computer science from Keio University, Tokyo, Japan, in 2009. He is currently a professor with Northwestern Polytechnical University. He was a postdoctoral researcher with the Institute TELECOM SudParis, Essonne, France. His current research interests include ubiquitous computing, social and community intelligence, mobile crowd sensing, and human-computer interaction. He is a senior member of the IEEE.