# How Big is the Crowd? Event and Location Based Population Modeling in Social Media

Yuan Liang, James Caverlee, Zhiyuan Cheng, Krishna Y. Kamath
Department of Computer Science and Engineering
Texas A&M University
{yliang, caverlee, zcheng, kykamath}@cse.tamu.edu

## ABSTRACT

In this paper, we address the challenge of modeling the size, duration, and temporal dynamics of short-lived crowds that manifest in social media. Successful population modeling for crowds is critical for many services including location recommendation, traffic prediction, and advertising. However, crowd modeling is challenging since 1) user-contributed data in social media is noisy and oftentimes incomplete, in the sense that users only reveal when they join a crowd through posts but not when they depart; and 2) the size of short-lived crowds typically changes rapidly, growing and shrinking in sharp bursts. Toward robust population modeling, we first propose a duration model to predict the time users spend in a particular crowd. We propose a time-evolving population model for estimating the number of people departing a crowd, which enables the prediction of the total population remaining in a crowd. Based on these population models, we further describe an approach that allows us to predict the number of posts generated from a crowd. We validate the crowd models through extensive experiments over 22 million geo-location based check-ins and 120,000 event-related tweets.

## 1. INTRODUCTION

Recently emerged social network services like Facebook, Twitter, and Foursquare are some of the largest and fastest growing web communities, offering an immense platform for connecting people together. Together, these services enable users to voluntarily enrich the physical world with their "footprints" – including geo-tagged photographs posted to Facebook, "check-ins" to location sharing services like Foursquare that link a user to a venue at a particular time, tweets in response to events and current affairs, and so on. These footprints provide exciting new opportunities for large-scale mining of the activities and patterns of millions of users, and promise to impact various areas including smarter emergency management applications [12], travel recommendation services [16], local news summarization [14] and regional advertising [8].

In one direction, these footprints can reveal the "crowding" behavior of large numbers of users. Example crowds range from users congregating at a particular place (e.g., a coffee shop), to users posting pictures and discussing a specific event (e.g., an earthquake), to users tweeting around a common interest (e.g., the Olympics), and so on. Tracking and characterizing these crowds as they arise and disappear is fundamental for many applications – including social media-based disaster management, where tweets can be mined to identify groups of affected users; enhanced traffic forecasting, where socially-augmented services like Waze provide evidence of emerging traffic jams; and political analysis, where tweets during significant political events can potentially reveal the collective mood.

Toward the goal of modeling crowds in social media, this paper focuses on modeling the *population dynamics* of these crowds as they first form, evolve, and eventually dissolve. Our goal is to study the potential of social media for building crowd population models that can estimate the dynamics, duration, periodicity, and life-cycle of crowds that may form in these systems. In this way, population models may reveal crowds that will continue to grow and those that are on the decline, as well as providing the basis for new advances. For example, robust population models built over user-contributed posts could predict future population density of restaurants, bars, and other local hotspots; urban planners and local governments could have access to real-time population maps, reflecting the current movements of people through space (rather than reflecting stale census estimates or relying on expensive sensors); companies and investors could adjust their marketing strategy, and allocate their limited resources based on the population of users drawing attention on their products; and political groups could estimate the percentage of people voting for or protesting against a new policy, with the help of population modeling for crowds.

Concretely, we focus our examination in this paper on two types of crowds:

- **Event-driven crowds:** reflecting a collection of people who are discussing or participating in a specific event, e.g., users posting about the superstorm Sandy or users participating in an anti-government protest in Syria.
- **Location-driven crowds:** reflecting groups who are bound to a certain place, e.g., people posting messages from Manhattan or from a Starbucks located in Pike Place, Seattle.

These two types of crowds provide a test case for develop-

*24th ACM Conference on Hypertext and Social Media*
*1–3 May 2013, Paris, France*

ing and validating generalized population models over social media. By considering crowding behavior in both, our hope is to reveal common patterns of crowding behavior. Concretely, we study a set of 22 million posts from location sharing services and a set of events containing 120k tweets and make the following contributions:

- First, we introduce the problem of modeling time-evolving populations for crowds in social media, identify the challenges of developing such models, and propose a general framework for estimating population from noisy and incomplete user-contributed posts.
- Second, we develop and evaluate models of user duration for location-driven and event-driven crowds, wherein we observe that durations for different crowds follow a power decay law and that semantically-correlated crowds display similar duration distributions.
- Third, we propose a novel population model incorporating user durations for estimating the number of people leaving from a crowd and hence predicting the total population remaining. The population model is validated through a traffic prediction scenario for Manhattan.
- Finally, based on the developed population models, we further propose and build an emission model for crowds, to model the artifacts generated by crowds (e.g., photos, posts). By incorporating the emission and population models, we show how to estimate the posts published by a crowd, which is important for predicting online traffic volumes for a certain event or service, detecting hotspot in social media, etc.

# 2. RELATED WORK

Recently, many efforts have focused on the temporal dynamics of online social networks and mobile networks. For example, [13] studied the periodicity of people's activities with respect to their most visited location. In [3] and [15], the authors showed that the daily and weekly check-in patterns for specific locations can reveal semantic information (e.g., that two locations are similar), and can facilitate location-based search and location recommendation. Researchers in [7] have explored the message number distribution between friends in the same school and different school [6] and in [1], the authors presented a study of the distribution of call duration in a mobile network and the contact duration in a Mobile Ad hoc Network (MANET).

Location sharing services have also attracted increasing attention in the last couple of years. [4] built models for human mobility patterns from 22 million posts from location sharing services, and studied different factors that affect people's mobility patterns. [18] examined users' histories of visited locations, and proposed a travel recommendation system by mining the correlation between users and their GPS trajectories. [5] analyzed the temporal, spatial and social dynamics of tweets during a fire emergency, and discussed how the location-based social network can be a source to collect information during emergencies. How and why people use location sharing services, especially Foursquare has been studied in [9].

# 3. PRELIMINARIES

In this section, we describe the basics of population modeling, highlight several challenges to successfully developing models over social media, and present the crowd-based datasets used in the following sections.

## 3.1 Challenges to Population Modeling

Intuitively, population can be modeled using the number of births, deaths, immigration, and emigration. In the basic population model, suppose a place has a population of $N_t$ at time $t$. Denoting the number of newborns as $B_t$, the number of deaths as $D_t$, the number of immigrants as $I_t$, and the number of emigrants as $E_t$, then the population for this place at time $t+1$ is defined as:

$$N_{t+1} = N_t + B_t - D_t + I_t - E_t \tag{1}$$

According to Equation 1, the population increase from times $t$ to $t+1$ is the difference between the number of births and deaths, plus the difference between the numbers of immigrants and emigrants. In the context of short-lived crowds, we can assume that the birth and death rates are close to 0, leading to a population model based purely on immigration and emigration:

$$N_{t+1} = N_t + I_t - E_t \tag{2}$$

Although seemingly straightforward, there are a number of challenges to model population from user-contributed artifacts in social systems:

- While it is natural to estimate immigration using check-in data observed from posts (reflecting users who newly joined a crowd), it is unclear how to estimate check-out behavior. Users typically do not explicitly indicate the time that they leave a crowd, meaning that population modeling via user-contributed posts alone is insufficient.
- Crowds in social media may suffer from data sparsity due to small coverage—since only a small percentage of people will post about their activities—and a low posting frequency—since the posts of a user may be too infrequent to capture fine-grained crowding behavior.
- Noise may also be introduced for many reasons including repetitive check-ins, incorrect location information, and misclassification of crowd-related posts.

## 3.2 Data Collection

For our analysis and experiments in the following sections, we use two sets of data: (i) a **Location Dataset** for analyzing location-driven crowds; and (ii) an **Event Dataset** for analyzing event-driven crowds.

**Location Dataset:** The first dataset contains user posts from several popular location-based services (e.g., Foursquare, Twitter, and Gowalla). Every post includes a timestamp (i.e., creation time) and the location (i.e., geographical coordinates) where the message was posted. This dataset was collected between October 2010 and January 2011 and it contains more than 22 million posts from 603,796 unique venues. About 62% of these posts are associated with a venue and every venue has about 23 posts on average [3].

**Event Dataset:** The second dataset contains event-related tweets that were collected using the Twitter API. Initially, we collected around 1 billion tweets from February 2011 to March 2012 with an average of around 3 million tweets every day. For our experiments using event-driven crowds we focus on 6 major events in the dataset, which are listed in Table 1.

To identify event-related tweets, we first determine a set of keywords associated with each event, and then keep only tweets passing an event similarity threshold. To determine the set of keywords associated with an event, we first identify one or two obvious keywords (like 'Irene' for Hurricane Irene). Using these seed keywords we identify 1,000 most co-occurring words, and then calculate their average term frequency (tf) per day during the time span of the event ($T1$) and during the ten days prior to the start of the event ($T2$), giving us $tf_{T1}$ and $tf_{T2}$. Then, we filter words that occur relatively infrequently during the event: $tf_{T1} < \lambda tf_{T2}$, where $\lambda$ is a threshold set to 3 in this case. This method yields about 20-50 event-related keywords, which are then used to represent an event as a vector. Next, the term vector for each tweet is computed by tokenizing the tweet using whitespace as a delimiter, and the cosine similarity of the keyword vector and the term vector is calculated. Finally, we keep all the tweets passing a similarity threshold in the dataset.

**Noise Filtering:** For the Location Dataset, the noises introduced by the incorrect location information are reduced by filtering out the check-ins from users whose successive check-ins imply a rate of speed faster than 1000 miles-per-hour [3]. For the Event Dataset, to reduce the noises caused by the misclassification of the event-related tweets, we keep only tweets within a pre-specified time frame and geographic boundary. For example, Linsanity had a timeframe of the first two weeks of February 2012 and was constrained to the geographic boundary of United States. We filter the tweets that are outside the time frame of the event or the corresponding boundary, and finally we get above 10,000 related tweets for each event (Table 1).

Table 1: Event Dataset

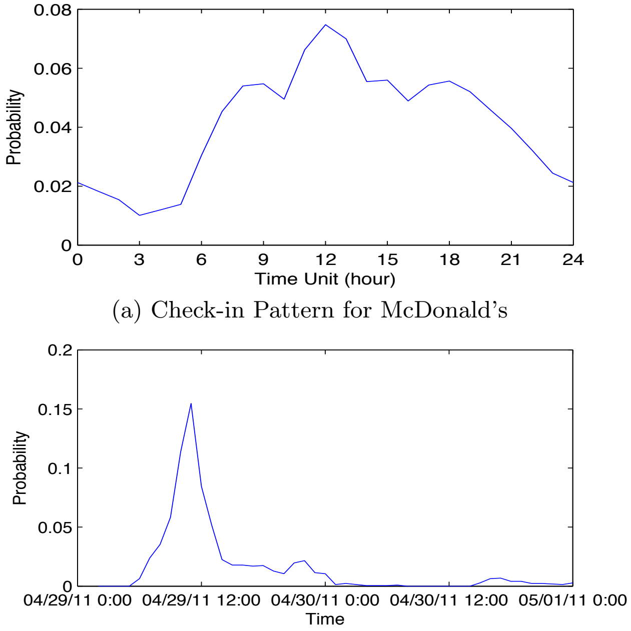
| Event | Period $T1$ | Box $B$ | Top3 words | #Tweet |
|---|---|---|---|---|
| JPEQ | 03/11/2011<br>-03/15/2011 | US | tsunami, japan,<br>earthquake | 19281 |
| Irene | 08/20/2011<br>-08/24/2011 | US | hurricane,<br>irene, tornado | 10352 |
| SteveJobs | 10/05/2011<br>- 10/09/2011 | US | steve jobs,<br>rip, apple | 31738 |
| Wedding | 04/29/2011<br>-05/03/2011 | UK | wedding,<br>royal, kate | 21551 |
| Linsanity | 02/04/2012<br>-02/14/2012 | US | jeremy lin,<br>linsanity, knicks | 10369 |
| Election | 04/01/2012<br>-04/30/2012 | US | obama, romney,<br>president | 30098 |

# 4. CROWD-BASED POPULATION MODEL

In this section, we develop the crowd-oriented population model. We show how to estimate the "immigration" (or check-in) population and the "emigration" (or check-out) population, before fully developing the population and emission-based models for understanding crowd dynamics.

## 4.1 Estimating "Immigration"

To estimate the number of "immigrants" for a crowd $r$, we model the temporal posting pattern for $r$ using the timestamps of user-contributed posts, and use this pattern as check-in pattern to approximate the actual population of people checking in at $r$ given a time.



(a) Check-in Pattern for McDonald's

(b) Check-in Pattern for Royal Wedding 2011

Figure 1: Examples of Check-in Patterns

Using the posts from the two data sets, we can generate check-in patterns for location-driven or event-driven crowds. For location-driven crowds, given a place $l$, we extract the timestamps of $l$'s posts, and normalize the timestamps into 24 time units representing the 24 hours in a day (if a user published multiple posts in $l$ within 24 hours, we only use her first post). An example check-in pattern for McDonald's is displayed in Figure 1a, where x-axis represents the time unit and y-axis is the normalized count of check-ins, representing the check-in probability given a certain time. We can clearly see that there are three peaks around 8:00, 12:00 and 18:00, and 12:00 has the highest frequency of check-ins. The peak times are consistent with breakfast, lunch and dinner time, and indicate that McDonald's is most popular for a quick lunch. For event-driven crowds, given an event $e$ and its period, we normalize the timestamps of associated posts per hour (for users who contribute multiple posts during the period of $e$, only the first one is taken into account). For example, Figure 1b is the check-in frequency of the Royal Wedding 2011 in the UK between April 29th and April 30th. We find that the check-ins burst at April 29, 2011, and the peak hour is around 11:00 local time, which is exactly the highlight of the whole wedding.

Similar check-in patterns have been studied in [3], [15] and [17], where the check-in patterns associated with locations or event were shown to reveal semantic information, for example for automatically grouping related locations based on the similarity of their check-in patterns (e.g., reflecting that coffee shops tend to have similar "immigration" patterns).

## 4.2 Modeling Duration to Estimate "Emigration"

In analogy to check-ins, we use checkouts to estimate the number of "emigrants" checking out from a specific crowd. However, since users only publish posts when they join a crowd – like tweeting when they arrive at a place or participate in an event, and do not explicitly post announcements when they leave – the checkout number cannot be measured directly. To solve this problem, we propose to model the duration of time that users spend in crowd $r$ to estimate when users will check out from $r$. The duration $d$ here refers to the time a user stays in crowd $r$. The probability of $d$ is formally defined in Equation 3, where $R$ is a crowd, and the subscript indicates the time period (e.g., the crowd at time $t$).

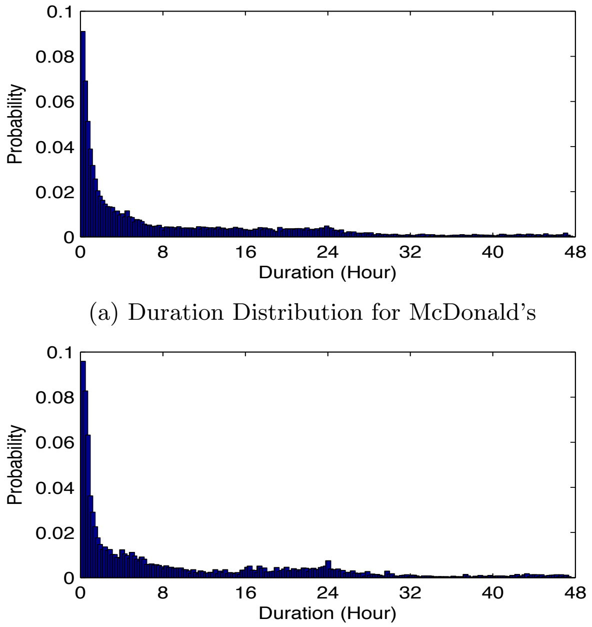
$$\begin{aligned} P(d|r) = P(&R_{t+d+1} \neq r, R_{t+d} = r, R_{t+d-1} = r, \\ &..., R_{t+2} = r | R_{t+1} = r, R_t \neq r) \end{aligned} \tag{3}$$

**Location-based durations:** For location-driven crowds, we assume that once someone checks in at location $l$, they will spend duration $d$ at location $l$. From a notation standpoint, the $r$ in Equation 3 can be replaced by $l$ to reflect a location-based crowd. Hence, we can estimate the duration $d$ given a location $l$ using the time span between every two posts from the same user, where the first post is from $l$ and the second post is the first one from a different location.

**Event-based durations:** Different from location-driven crowds where a user can only be in one crowd at a specific time $t$, event-driven crowds may attract participants who express interest in multiple events since there is no physical requirement of being present at a particular location. As a result, these users can follow multiple events at the same time, and they may leave and return to a particular event $e$ over time. Therefore, we estimate the duration for event-driven crowds in a different way: from the posts related to an event, we find all the posts $R_{t_1}, R_{t_2}, ..., R_{t_n}$ ($t_i$ is the index of posts) that belong to a user $u$, and then use the interval of her first and last post as the duration $d$. $t_i (i = 1, 2.., n)$ do not need to be successive (like the posts for location-driven crowds do). If there is only one tweet for a user in $e$, then we assume the duration is 0, which means this user does not stay in $e$.

*Example:* Taking the crowds at McDonald's and Best Buy as examples, the duration distributions for these location-driven crowds are plotted in Figure 2. To illustrate event-driven crowds, the duration distributions for the people involved in the Japan earthquake and the Royal Wedding are shown in Figure 3.

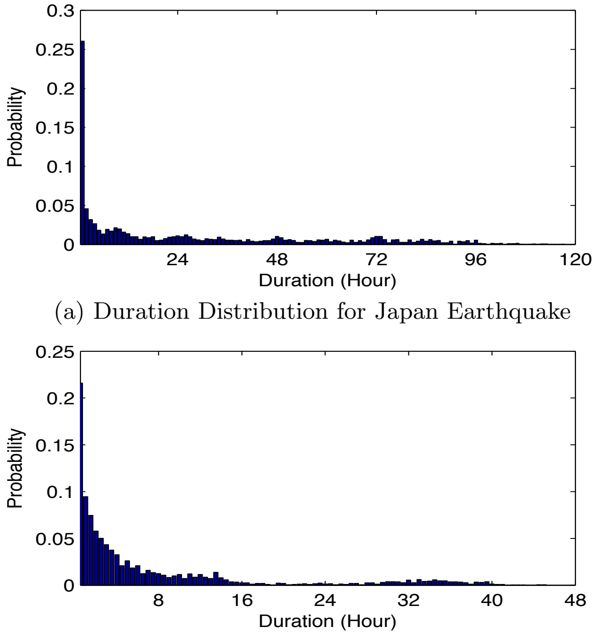
For McDonald's and Best Buy, the probabilities peak in the first half an hour and decrease following a power decay law when the duration increases, which appears intuitively reasonable. However, we also observe that the durations derived from measuring inter-posts times display some anomalies. Since many users may post only infrequently, we see that there are a number of people apparently with a duration of 24 hours or more at McDonald's. Similarly, we see a spike after 24 hours at McDonald's and Best Buy, most likely capturing people who posts only for these location and nowhere else (resulting in a one day "duration").



(a) Duration Distribution for McDonald's

(b) Duration Distribution for Best Buy

Figure 2: Example of Duration Distributions for Location-Driven Crowds



(a) Duration Distribution for Japan Earthquake

(b) Duration Distribution for Royal Wedding

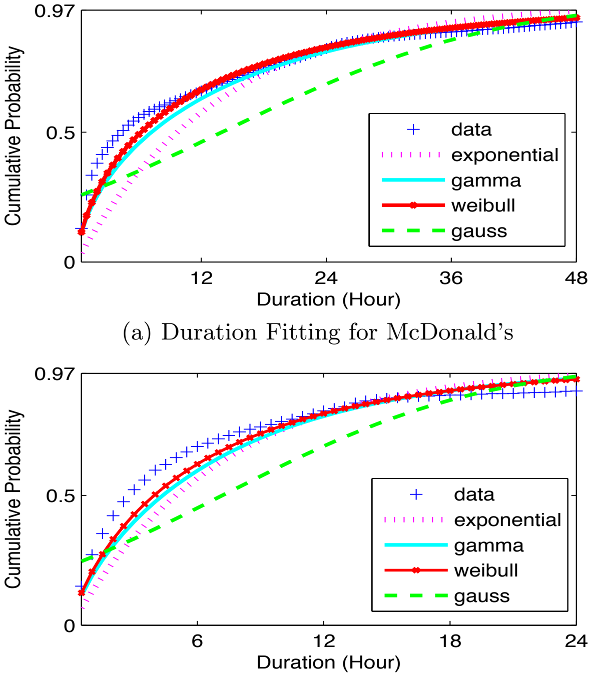
Figure 3: Examples of Duration Distributions for Event-Driven Crowds

Figure 3 shows that the duration probability for event-driven crowds also follows a power decay law and has a long tail (Figure 3 only plots the duration distribution of the users who spend $d > 0$ on the events). Compared with the location-driven crowds, people tend to spend less time on events. For example, the probability $P(d = 0|JPEQ)$ is 77.61%, whereas $P(d = 0|Wedding)$ is 56.72%, which means that fewer users stay associated with an event. That is consistent with the reality that most people are just transient viewers for an event but not long-term participants. And comparing to location-driven crowds, they have a longer tail, because events usually last longer.

### 4.2.1 Duration Distribution Fitness

To better understand the duration distribution, given a crowd $r$, we next examine a series of distributions which are commonly used for duration modeling. In different applications, different probability density functions have been adopted for duration modeling, e.g. [1] proved that the contact duration follows an Exponential pdf in a mobile ad-hoc network, [10] used a Weibull pdf to estimate the response time for traffic incidents. Here, we consider four alternatives: Gaussian, Exponential[1], Gamma[10], and Weibull[2].

Taking two crowds as examples – McDonald's and the Royal Wedding – we fit their cumulative distributions of duration using different pdfs, and illustrate the best-fit results in Figure 4.



(a) Duration Fitting for McDonald's

(b) Duration Fitting for Royal Wedding

Figure 4: Duration Cumulative Distribution Fitness

In Figure 4, we observe that the Weibull pdf fits the data best, followed by the Gamma pdf, and then the Exponential pdf. The Gaussian pdf achieves the worst fitness. Applying the Kolmogorov-Smirnov test also indicates that the Weibull pdf fits the data better than other possibilities. Usually Weibull is more flexible than the Exponential distribution in which the probability of users staying an additional period may depend on their current duration. Therefore it's quite suitable for modeling the duration for these two crowds.

## 4.3 Building the Population Model

Given the duration model defined in the previous section, we now propose a time-evolving population model that estimates both when users will depart a crowd and how many users are remaining in the crowd.

Given a crowd $r$ and a time $t$, the number of people who will check out at time $t$ is the sum of the number of people who checked in at $t_0$ and stay at the location for $t - t_0$ and the people who checked in at $t_1$ and stay there for $t - t_1$ and so on. The people who checked in $d$ hours ago before time $t$ check out with probability $P(d|r)$. So denoted $Q_{out}(t|r)$, the population checking out from $r$ at $t$ is in Equation 4, where $Q_{in}(t|r)$ is the check-in population given a crowd $r$ at time $t$, and $P(d|r)$ is the duration probability for people to stay in $r$ for duration $d$.

$$Q_{out}(t|r) = \int_{t'=0}^{t} Q_{in}(t'|r)P(t-t'|r)\mathrm{d}t' \qquad (4)$$

Equation 4 is a convolution of check-in function and duration functions, the effect is that the volume of check-ins is smoothed and shifted backward by the duration function.

Given a crowd $r$ and a timestamp $t$, the remaining population is the difference of the total number of people who have checked in before $t$ and the total number of people who have checked out before $t$. And people who checked in $d$ hours ago before time $t$ would remain with the probability $1 - C(d|r)$, where $C(d|r)$ is the cumulative distribution function for $d$. Therefore, we can estimate the population remaining in $r$ at $t$, denoted by $Q_{rem}(t|r)$, as:

$$Q_{rem}(t|r) = \int_{t'=0}^{t} Q_{in}(t'|r)(1 - \int_{d=0}^{t-t'} P(d|r)\mathrm{d}d)\mathrm{d}t' \qquad (5)$$

where $\int_{d=0}^{t-t'} P(d|r)\mathrm{d}d$ is the cumulative probability $C(t - t'|r)$.

In our experiments described in Section 5, given a crowd $r$, a series of distributions including Gaussian, Exponential, Gamma and Weibull will be tried for $P(d|r)$.

## 4.4 Emission-Based Modeling

Based on the population model, we can further estimate the emissions of the crowds. An *emission* here refers to the products that a user "emits" during their stay in a crowd. To illustrate, for a crowd of people attending the 2012 London Olympics, some will "emit" videos and photos by uploading them to social media sites like Facebook. For a crowd of Apple iPhone 5 fans, some will actually purchase the iPhone, resulting in a crowd emission. For tourists visiting Manhattan, we could view their associated traffic volume as an involuntary crowd-based emission.

Our goal in this section is to develop an emission model for capturing these crowd-based products. Depending on the application domain, the emission model could be useful across a number of settings including web service providers, product review systems, and hotspot detection applications. Concretely, we focus our model on the tweets emitted by a crowd based on our event-based crowd population model.

To estimate the number of tweets, we modify Equation 4 to Equation 6 by considering the post count per user when they stay in crowd $r$. Given a time $t$, the users staying in $r$ are those whose check-in time $t_s \leq t$ and checkout time $t_e > t$ (their duration is $d = t_e - t_s$). The number of those people can be estimated with $Q(t|r) = \int_{t_s=0}^{t} \int_{t_e=t}^{\infty} Q_{in}(t_s|r)P(t_e - t_s|r)\mathrm{d}t_s\mathrm{d}t_e$. And for each user, the expected number of the posts is $\lambda = \sum_{n=1}^{MaxN} n\eta(n)$, where $n$ is the number of posts, $\eta(n)$ is the probability mass function (pmf) for $n$. And given a time $t \in [t_s, t_e)$, let the probability that a user posts an annotation at $t$ is $\gamma(t|t_s, t_e)$. Then the number of posts at $t$ can be computed with the user count $Q(t|r)$ at $t$ times the posts count per user $\lambda\gamma(t|t_s, t_e)$ at $t$. The formula is shown in Equation 6:

$$Q_{post}(t|r) = \lambda \int_{t_s=0}^{t} \int_{t_e=t}^{\infty} Q_{in}(t_s|r)P(t_e - t_s|r)\gamma(t|t_s, t_e)\mathrm{d}t_s\mathrm{d}t_e \qquad (6)$$

We model the posting probability function $\gamma(t|t_s, t_e)$ with three distributions: 1) Uniform distribution; 2) Exponential distribution; and 3) U-shaped distribution. We use these functions to check whether the posts of users are evenly distributed or concentrated on the beginning or ending during the event duration $[t_s, t_e)$.

**Uniform distribution:** In this function, we assume that during period $[t_s, t_e)$, each time point has the same probability to emit a post.

$$\gamma(t|t_s, t_e) = \begin{cases} \frac{1}{t_e - t_s} & t_s \le t < t_e \\ 0 & else \end{cases}$$

**Exponential distribution:** In this approach, we believe it is more likely for people to post when they just check in, and then the chance for posting decreases exponentially when time passes. In the following equation, $\alpha = \int_{t=t_s}^{t_e} e^{-(t-t_s)} dt$ is the normalizing factor.

$$\gamma(t|t_s, t_e) = \begin{cases} \alpha e^{-(t-t_s)} & t_s \le t < t_e \\ 0 & else \end{cases}$$

**U-shaped distribution:** In this function, we assume that people tend to post when they check in and check out, so the chance is high at their check-in time, then decreases exponentially until at the middle of the period $[t_s, t_e)$, then the chances increase exponentially until at the checkout time. It is a combination of two Exponential function, constituting a U-shape probability function.

$$\gamma(t|t_s, t_e) = \begin{cases} \alpha e^{-(t-t_s)} & t_s \le t < \frac{t_s+t_e}{2} \\ \alpha e^{-(t_e-t)} & \frac{t_s+t_e}{2} \le t < t_e \\ 0 & else \end{cases}$$

where $\alpha = \int_{t=t_s}^{\frac{t_s+t_e}{2}} e^{-(t-t_s)} dt + \int_{t=\frac{t_s+t_e}{2}}^{t_e} e^{-(t_e-t)} dt$ is the normalizing factors for the function.

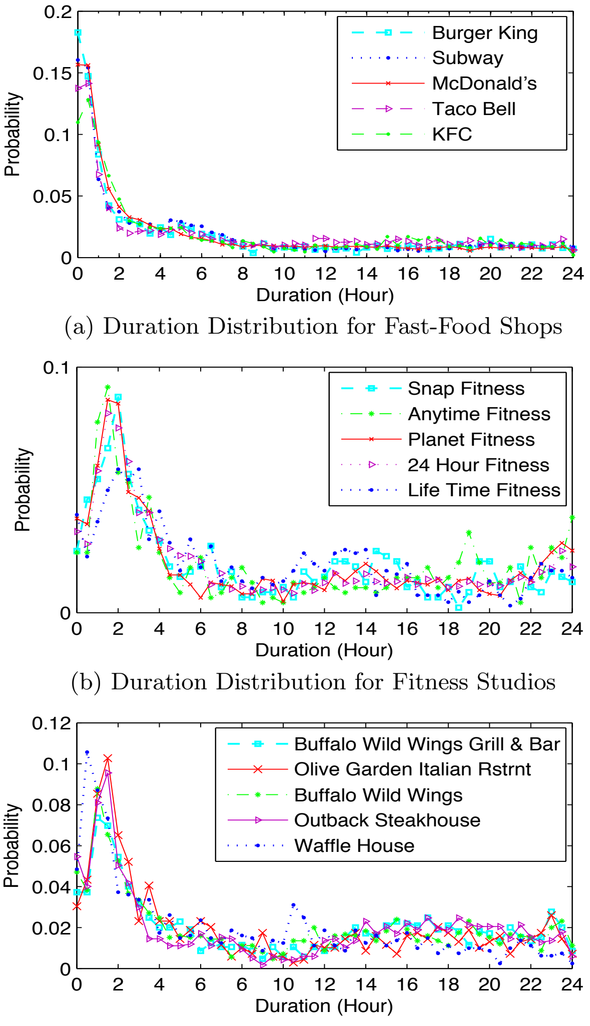
# 5. EXPERIMENTS

We design three sets of experiments to verify the proposed duration, population and emission models respectively.

- In the first set of experiments, our goal is to analyze duration and determine if it is informative. We are interested in evaluating if for a given venue the duration patterns modeled with data reflect the actual time users spend in the venue.
- In the second set of experiments, we analyze if the population model described in this paper can be used to predict traffic volume. To verify this we use the traffic information on Manhattan's bridges as an example. Given the incoming traffic volume to Manhattan, we first train the duration model for Manhattan using Location Dataset. We then estimate the checkout volume using proposed population models and compare it with the actual outgoing volume to verify these models.
- In the third set of experiments, we verify the population and emission models with respect to event-driven crowds. We evaluate if for a given event and its posts, the population model can estimate the checkouts from that event and the emission model can accurately predict actual number of posts written by the crowd corresponding to that event.

## 5.1 Is Duration Informative?

Even though the duration as measured through inter-post gaps is clearly noisy and not immediately informative, perhaps there are interesting patterns in this duration distribution across crowds? Examining the duration patterns for venues of different types, we plot in Figure 5 the duration distributions for three categories of venues: fast food restaurants, fitness centers, and casual restaurants. To reduce the noise of incorrect large durations caused by infrequent posts, we remove all the $d$ with $d > \eta$, where $\eta$ is set with 24 hours here. Interestingly, we observe that the duration distribution agrees with our expectation that venues in the same category share very similar patterns. Across different categories, the duration patterns of retail stores and fast-food restaurants are dramatically different from the ones of fitness centers and restaurants, indicating that people tend to spend less time on fast-food venues and retail stores than fitness centers and restaurants.



(a) Duration Distribution for Fast-Food Shops

(b) Duration Distribution for Fitness Studios

(c) Duration Distribution for Restaurants

Figure 5: Comparison of Duration Distributions

To further investigate whether the duration pattern can reflect the actual time users spend in a location, we check the duration patterns of venues of different types and propose to analyze the semantic correlation between them, by grouping related venues based purely on check-in and derived durations revealed through location sharing services. We believe if the duration model suggests the real pattern of users' behaviors in physical world, we can use it as a feature for semantic analysis of locations.

We sample 114 venues with the largest number of posts (posts of all distinct venues owning the same name are aggregated, e.g., combining all distinct Starbucks into a single "Starbucks" location) and retrieve their features including: check-in feature in the form of 24 dimension vector $(c_1, c_2, ..., c_{24})$, the $i^{th}(1 \leq i \leq 24)$ dimension in the vector stands for the probability of check-ins at $i^{th}$ hour during 24 hours; and a duration feature containing 24 dimensions $(d_1, d_2, ..., d_{24})$, the $d^{th}(1 \leq d \leq 24)$ component is the probability of duration is equal to $d$ hours.

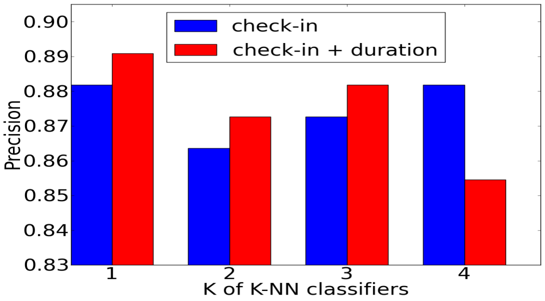


Figure 6: Venue Classification

Given the semantic category information ("Food", "Shop", or "Home, Work and Other") retrieved from Foursquare for each of the 114 venues, we consider the set of 114 venues as ground truth data, and we consider the distribution of duration as another feature in addition to the check-in pattern to predict the category label for the venues. We apply a kNN classifier (using Euclidean distance as similarity measure between venues) on the set of 114 venues, and use 10-fold cross validation to evaluate whether the use of duration distribution can improve the identification of semantically-related venues. The classification results for kNN (k = 1, 2, 3, 4) is shown in Figure 6. As we can see from the figure, augmenting a baseline classifier with duration information yields positive results in most cases, suggesting that duration is an informative characteristic derived from location sharing services.

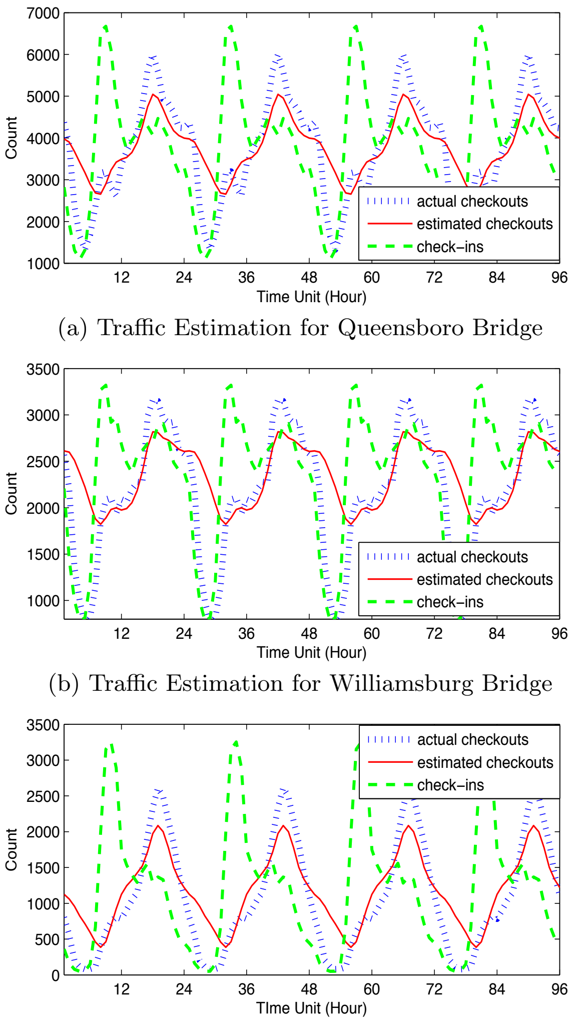
## 5.2 Traffic Prediction with Population Model

To verify the population model, we propose to use it to predict traffic conditions. With this goal, we need to build a traffic prediction model. Simply, we treat an area as an enclosed box with only several outlets, whereby people check in and check out using these outlets. Suppose we already know the check-ins for this area, using Equation 4, then the checkout population can be estimated. We take Manhattan as an example, which is an enclosed area, the bridges and tunnels connect Manhattan and the surrounding districts including New Jersey, Brooklyn, Queens, Bronx. The traffic volumes on these bridges and tunnels from the surrounding districts to Manhattan are treated as check-ins, and the traffic volumes from Manhattan to these areas are treated as checkouts. We consider 19 two-way bridges and tunnels, e.g., the Manhattan Bridge, Brooklyn Bridge, Queensboro Bridge, Williamsburg Bridge, Gorge Washington Bridge, Holland Tunnel, Lincoln Tunnel, Washington Bridge, Alexander Hamilton Bridge, and so on.

The traffic volume data comes from the report "New York City Bridge Traffic Volumes 2010" [11]. It lists the average hourly traffic volumes importing to and exporting from Manhattan through the bridges in 2010, which are two 24-hour volume vector $(in_1\ , in_2\ , \ ... \ in_{24})$ and $(out_1,\ out_2,\ ...\ out_{24})$. We use $(in_1,\ in_2,\ ...\ in_{24})$ as check-in data, and $(out_1,\ out_2,$ $...\ out_{24})$ as the ground truth. Equation 4 is used to estimate the number of check-outs. The duration pdf $P(d|Manhattan)$ in Equation 4 describing how long people stay in Manhattan is trained with the data set containing 22 millions geo-located based posts. A duration is computed by considering the interval of two successive posts by the same user in the Manhattan (and neighboring) areas. For example, if a user checks in at Brooklyn at $t_1$, then he posts in Manhattan at $t_2, t_3, ..., t_{n-1}$, and then checks in at Brooklyn again at $t_n$, we consider $t_n - t_1$ as a duration. With the prior knowledge that most traffic volume come from commuters shuttling between two districts, we filter the durations larger than 12 hours. At last we collect 1,673 durations to train $P(d|Manhattan)$. Both discrete and continuous functions including Gaussian, Gamma, Exponential and Weibull pdfs are experimented for $P(d|Manhattan)$.

In Figure 7, the green dotted line is the repeat check-in frequencies of 4 days, the red solid line is the estimated checkout population using the incoming volumes and the $P(d|Manhattan)$, and the blue dotted line is the actual outgoing volume (checkouts). To illustrate, Figure 7a shows the check-ins and checkouts for the Queensboro Bridge. The peak time of check-ins is about 8 am, the peak time of actual checkouts is about 5 pm, which is intuitively consistent with habits arranged around a work schedule. The estimated checkouts display a similar trend with the actual ones and also peak at 5 pm. In Figure 7b and 7c, the estimated checkouts also fit the ground truth well. This suggests that the duration distribution can correctly capture the lag between check-ins and checkouts, and that the population modeling method with duration is effective in estimating the size of crowds, and correctly capturing the dynamics in population with time passing. We also notice that the estimated checkouts do not exactly fit the actual checkouts. We attribute this difference to three main reasons: 1) in real traffic, not all incoming vehicles will also depart (check out) by the same route; 2) outgoing volumes may also be contributed by other crowds which are not shuttling between Manhattan and its four neighbor districts; and 3) incomplete coverage of users' trajectory may lead to inaccurate estimation of their duration.

Though the exact traffic volume is hard to predict, we can use the estimated checkouts to predict the relative volumes. So we rank the time units according to the number of checkouts, and compare the ranked list with the list ranked with actual checkout volumes. Thus the prediction problem may be viewed as a rank problem. And since the top ranked time units (rush hours) are what we are concerned with most, therefore we can use NDCG (normalized DCG - discounted cumulative gain) as the metric to evaluate the ranked results. The equation of NDCG is given in Equation 7, where the $rel_i$ is the score for a time unit, IDCG - ideal DCG - is the maximum possible DCG till position k.



(a) Traffic Estimation for Queensboro Bridge

(b) Traffic Estimation for Williamsburg Bridge

(c) Traffic Estimation for Brooklyn Battery Tunnel

Figure 7: Traffic Prediction of Bridges of Manhattan

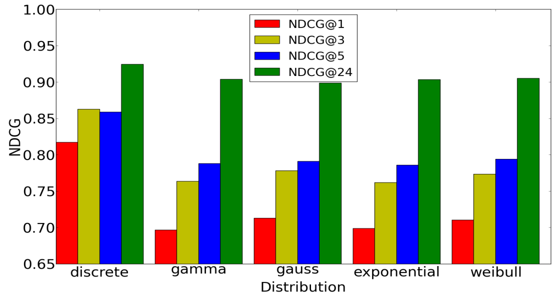


Figure 8: NDCG for The Rush Hours Ranked by the Estimated Number of Checkouts

To calculate $rel_i$, we first compute the number of checkouts $Q_{out}(t|Manhattan)$ for 24 time units using our models, then we rank the units decreasingly with $Q_{out}(t|Manhattan)$ and assign each unit a score according to its rank, e.g. the 1st unit is given 24, 2nd is given 23. The ranked results are shown in Figure 8.

$$NDCG@k = \frac{DCG}{IDCG} = \frac{rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2 i}}{IDCG} \qquad (7)$$

For different NDCG@k, our population models all achieve above 70% gains, especially the model using discrete duration distribution reach above 80% gains for all $k$s. This indicates that the checkout trends can be well estimated with the proposed population model. The different continuous duration achieves comparable performances, where generally Weibull slightly outperforms the other distributions, which is consistent with the hypothesis test result in Section 4.2.1. The discrete duration distribution is better than the continuous ones; one reason is the limited data undertrains the models, and the trained continuous model oversmooths the checkouts.

## 5.3 User and Post Prediction with Population Model

In this part, two sets of experiments are conducted to verify the population model and emission model respectively. In the first experiment, we use the population model to estimate the checkouts for the event-driven crowds. Given an event $e$, the ground truth of checkouts $Q_u(t|e)$ are collected using the number of users who publish a post about $e$ at $t$ and never tweet about $e$ after $t$. The input data check-ins $Q_{in}(t|e)$ are collected with the number of new users per hour. The $P(d|e)$ is trained with all the pairs of two successive posts related to event $e$ of the same user. Equation 4 is used to estimate the number of users who check out $Q_{out}(t|e)$ at time $t$.

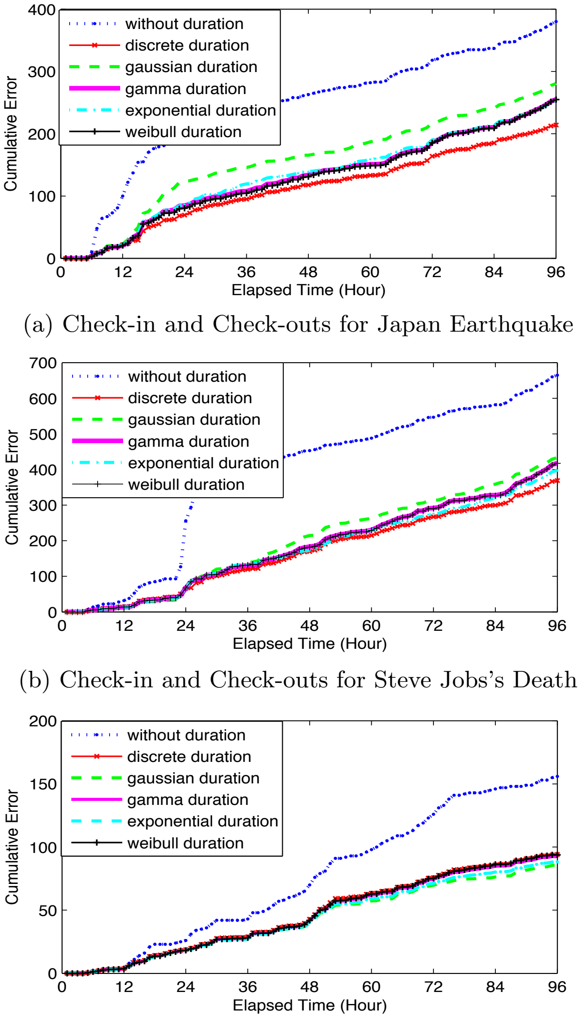
In the second experiment, we apply the emission model to estimate the number of posts for events. Given an event $e$, we collect $Q_w(t|e)$ - the number of tweets about $e$ per hour as the ground truth and check-ins $Q_{in}(t|e)$ as the inputs. Equation 6 is used to calculate the emitting posts. In Equation 6, the expected posts number $\lambda$ and $P(d|t)$ are trained with Event Dataset. To verify the emission model, we compare our estimated post number with the actual post number $Q_w(t|e)$, and residual sum of squares (RSS) is used to evaluate the results.

In both of the experiments, for each event, we randomly divide the tweets into two parts: training data containing the tweets of 80% users and testing data containing the tweets of 20% users.

In Figure 9 and Figure 10, we show the cumulative errors between the estimated checkouts with the actual checkouts. We compare our models with the method with no durations. The cumulative error is calculated using Equation 8.

$$Error(t) = \sum_{t'=0}^{t} |Q_{est}(t'|e) - Q_{real}(t'|e)| \qquad (8)$$

In Figure 9, the blue dotted line is the error between the actual checkouts and the estimated checkouts using the method with no duration (without duration, the number of checkouts is the same with that of check-ins). In all three examples in Figure 9, our models outperform the method with no duration. Among the models with different duration pdfs, the Gaussian pdf achieve the worst results in the short-term events, and approaches a good result in the long-term event. Weibull, Gamma and Exponential pdfs have very similar performance, and averagely they achieve the

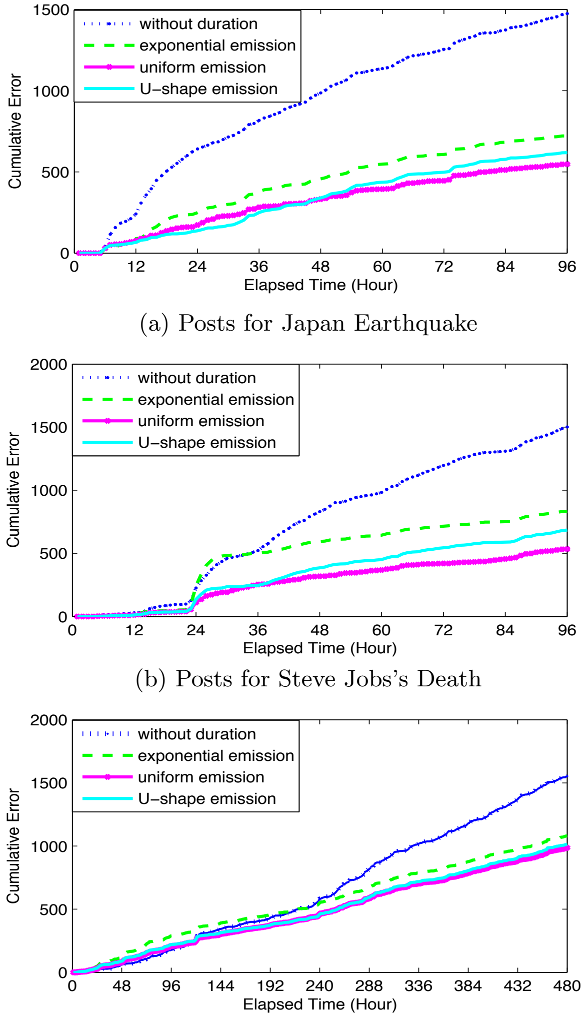


(a) Check-in and Check-outs for Japan Earthquake

(b) Check-in and Check-outs for Steve Jobs's Death

(c) Check-in and Check-outs for US President Election

Figure 9: Estimating Check-outs for Events. The X axis is the elapsed time since the start time of events.



(a) Posts for Japan Earthquake

(b) Posts for Steve Jobs's Death

(c) Posts for US President Election

Figure 10: Estimating Event-Related Posts. The X axis is the elapsed time since the start time of events.

better results than Gaussian distribution. This fact again validates the analysis in Section 4.2.1. The discrete function performs the best in the short-term events, probably because that limited training data set does not perfectly display the properties of given pdfs, leading to that all the continuous pdfs do not well fit the data set.

Figure 10 shows that cumulative error for the estimated count of posts for different events. Based on the previous experimental results, we adopt the discrete duration distribution for estimating the post count. In Figure 10, the blue dotted line is the cumulative error between the actual counts of posts and the estimated ones of posts using the method with no duration (without duration, the posts are the ones contributed by check-in users). In all three examples, our models are better than that with no duration. And among different emission distributions, the Uniform pdf achieves the best performance, Exponential pdf achieves the worst performance. This result indicates that for the people who would like to stay in an event (most users do not stay in an event, their duration is 0), they tend not to write all their posts at the beginning, neither only post at the beginning and end time of the event, in fact they are more likely to evenly distribute their annotations.

Next, we list the RSS for the estimated number of checkout users and posts in Table 2 and 3. Table 2 shows that the root RSS of estimated checkouts and posts using our models are much smaller than those of the method without considering duration. For each data set, our best model reduce the root RSS by 50.91%, -16.94%, 64.29%, 50.19%, 50.30% and 10.15% respectively. Among all the duration distributions, discrete function achieves the best performance. Exponential, Gamma, Weibull distribution approach very similar results, Gauss pdf perform worst averagely.

Table 3 shows that for each event, our best model reduce the root RSS by 59.55%, 26.66%, 56.03%, 77.97%, 60.00% and 37.05% respectively. Generally, Uniform distribution gets the best results, while for event Irene and Linsanity, the U-shape distribution is the best. The two events have a similarity that they both have multiple bursts, since Irene hurricane is a swift disaster and Linsanity erupted every time Knicks wins. Users' comments tend to concentrate in these bursts, and U-shape pdf has two bursts, so it might fit

Table 2: Estimating Checkout User Count

| event | no dura. | Discrete | Exp. | Gamma | Gauss | Weibull |
|---|---|---|---|---|---|---|
| JPEQ | 68.000 | 33.382 | 40.436 | 40.411 | 46.033 | 39.963 |
| Irene | 32.665 | 38.199 | 39.814 | 40.767 | 41.863 | 40.619 |
| SteveJobs | 165.360 | 59.049 | 66.456 | 70.199 | 71.288 | 70.361 |
| Wedding | 169.505 | 84.436 | 103.266 | 100.767 | 121.192 | 98.166 |
| Linsanity | 56.665 | 28.165 | 29.143 | 28.917 | 28.948 | 29.008 |
| Election | 47.138 | 42.355 | 43.379 | 44.206 | 44.935 | 44.186 |

the data better than other pdfs. And for royal wedding, the Exponential performs the best, this might be due to the fact that the royal wedding is the shortest-term event in these events. People are likely to talk about the event intensively right at the wedding procession, but will not look back later after the wedding.

Table 3: Estimating Post Count

| event | no dura. | Uniform | Exponential | U-shape |
|---|---|---|---|---|
| JPEQ | 200.242 | 80.997 | 102.046 | 88.372 |
| Irene | 61.073 | 46.811 | 59.630 | 44.790 |
| Steve Jobs | 218.958 | 96.285 | 211.488 | 127.921 |
| Wedding | 862.541 | 293.299 | 190.059 | 335.976 |
| Linsanity | 121.737 | 48.691 | 74.615 | 43.432 |
| Election | 117.694 | 74.089 | 80.998 | 78.141 |

In conclusion, though our training data is very noisy and sparse, the duration model is informative for the location-driven crowds. The proposed population model based on duration modeling is effective in estimating the checkout population for both location-driven and event-driven crowds. And the emission model works well on estimating the post number for event-driven crowds.

# 6. CONCLUSIONS

In this paper, we propose to model population given user-contributed posts for crowds. Opportunity for and challenges to population modeling in this noisy and incomplete domain are examined, and a novel time-evolving population model is proposed. To model the population, we investigate the distribution of duration derived from posting data, and we observe that the durations for location-driven and event-driven crowds follow an power decay law. In addition, given the check-ins and duration models, we are able to estimate the checkout population for crowds at a specific time. We further apply the population models to emission modeling for crowds to estimate the volume of their posts. Finally, we apply the population model to the traffic prediction and event-driven crowds' population estimation problems. Evaluation with the examples of Manhattan and a set of events shows effectiveness of the proposed models.

# 7. ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-1149383, DARPA grant N66001-10-1-4044 and a Google Research Award. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

# 8. REFERENCES

[1] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy. Modeling path duration distributions in manets and their impact on reactive routing protocols. *IEEE Journal on Selected Areas in Communications*, 22(7), 2004.

[2] R. J. Butler and J. D. Worrall. Gamma Duration Models with Heterogeneity. *The Review of Economics and Statistics*, 73(1):85–102, 1998.

[3] Z. Cheng, J. Caverlee, K. Kamath, and K. Lee. Toward Traffic-Driven Location-Based Web Search. In *CIKM*, 2011.

[4] Z. Cheng, J. Caverlee, and K. Lee. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

[5] B. De Longueville, R. Smith, and G. Luraschi. OMG, from here, I can see the flames!: A Use Case of Mining Location Based Social Networks to Acquire Spatio-temporal Data on Forest Fires. In *Workshop on Location Based Social Networks*, 2009.

[6] P. O. V. De Melo, L. Akoglu, C. Faloutsos, and A. A. Loureiro. Surprising Patterns for the Call Duration Distribution of Mobile Phone Users. In *ECML PKDD*, 2010.

[7] S. A. Golder, D. M. Wilkinson, and B. A. Huberman. Rhythms of social interaction: Messaging within a massive online network. In *the Third Communities and Technologies Conference*, 2007.

[8] P. Kitano and K. Boer. The local business owner's guide to twitter, 2009.

[9] J. Lindqvist, J. Cranshaw, J. Wiese, J. Hong, and J. Zimmerman. I'm the mayor of my house: Examining why people use foursquare - a social-driven location sharing application. In *SIGCHI*, 2011.

[10] D. Nam and F. Mannering. An exploratory hazard-based analysis of highway incident duration. *Transportation Research Part A: Policy and Practice*, 34:161–166, 1991.

[11] J. Sadik-Khan. New York City Bridge Traffic Volumes 2010, 2010.

[12] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[13] C. Song, Z. Qu, N. Blumm, and A.-L. Baraba´si. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[14] S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In *ICWSM*, 2010.

[15] M. Ye, D. Shou, W. C. Lee, P. Yin, and K. Janowicz. On the semantic annotation of places in location-based social networks. In *17th ACM SIGKDD*, 2011.

[16] M. Ye, P. Yin, and W. C. Lee. Location recommendation for location-based social networks. In *SIGSPATIAL*, 2010.

[17] M. Y. E. Zhang, H. Korayem and C. D.J. Beyond Co-occurrence: Discovering and Visualizing Tag Relationships from Geo-spatial and Temporal Similarities. In *WSDM*, 2012.

[18] Y. Zheng, X. Xie, and W. Ma. GeoLife: A Collaborative Social Networking Service among User, location and trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–34, 2010.