GeoSocialBound: An Efficient Framework for Estimating Social POI Boundaries Using Spatio–Textual Information

Dung D. Vu* Hien To+ Won-Yong Shin* Cyrus Shahabi+
*Computer Science and Engineering, Dankook University
+Integrated Media Systems Center, University of Southern California
{dungdovu,wyshin}@dankook.ac.kr
{hto,shahabi}@usc.edu

ABSTRACT

In this paper, we present a novel framework for estimating *social point-of-interest (POI)* boundaries, also termed *GeoSocialBound*, utilizing spatio–textual information based on geo-tagged tweets. We first start by defining a social POI boundary as one small-scale cluster containing its POI center, geographically formed with a convex polygon. Motivated by an insightful observation with regard to estimation accuracy, we formulate a constrained optimization problem, in which we are interested in finding the radius of a circle such that a newly defined objective function is maximized. To solve this problem, we introduce an efficient optimal estimation algorithm whose runtime complexity is *linear* in the number of geo-tags in a dataset. In addition, we empirically evaluate the estimation performance of our GeoSocialBound algorithm for various environments and validate the complexity analysis. As a result, vital information on how to obtain real-world GeoSocialBounds with a high degree of accuracy is provided.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral sciences

General Terms

Algorithms, Human Factors, Measurement

Keywords

$\mathcal{F}$-measure, Geographic Distance, Geo-Tagged Tweet, Social Point-of-Interest (POI) Boundary, Spatio–Textual Information, Twitter

1. INTRODUCTION

Location-based social networks (LBSNs) such as Foursquare and Flickr have grown rapidly in recent years. They provide a platform for millions of users to share their location-tagged media contents such as photos, videos, musics, and texts. Owing to the location information from geo-tags, there has been a steady push to study a variety of point-of-interest (POI) issues [1–3] through LBSNs. In general, when users visit a POI, they are likely to check in online or post photos of their visit via LBSNs to describe that their locations are related to the POI. Since the relevance of the data to the POI varies according to the geographic distance between the POI and the locations where the data are generated, it is important to characterize an area-of-interest (AOI) [4–6] that enables to utilize the location information in a variety of business advertisements.

Instead of geo-tagged photos collected from LBSNs, we utilize geo-tagged tweets on Twitter [7–11], which is one of the most popular micro-blogs holding a substantial amount of user accounts and records. In this paper, by using spatio–textual information on Twitter, we characterize a novel framework for estimating a *social POI boundary*, also termed *GeoSocialBound*. A social POI boundary is defined as only one *small-scale* cluster containing its POI center and is geographically formed with a convex polygon. Given such a social POI boundary, we can envision a broad range of useful applications including but not limited to:

- Location advertisements: As a marketing strategy of companies aimed at a POI (e.g., a shopping mall), leaflets/brochures online will be disseminated only to the people who come to visit the place. Thus, company managers will not only be aware of the explicit marketing zone but also reduce the marketing cost.
- Event management: A recommendation of good positions within a social POI boundary will be possible when companies want to launch their events near the POI.
- Traffic control: When there is the increase in traffic volume above the road capacity (e.g., a festival at a POI), traffic congestion will be significantly reduced by recommending the best route based on a social POI boundary for festival participants.

To estimate a social POI boundary with a high degree of accuracy, we explore the $\mathcal{F}$-measure, which has been accepted as the most important score to judge a boundary. We observe that the $\mathcal{F}$-measure, denoted by $\mathcal{F}(c, r)$, tends to be hardly degraded up to a certain critical point with respect to the radius r from the POI center c. Then, we formulate a new constrained optimization problem, in which we are interested in finding the radius r such that the objective function expressed as the *product of a power law in*

GeoRich'16, June 26, 2016, San Francisco, CA, USA

r and $\mathcal{F}(c,r)$ is maximized. To solve this problem, we propose an efficient GeoSocialBound algorithm whose runtime complexity is *linear* in the number of geo-tags in a dataset. Our main contributions are fourfold as follows:

- We first characterize a social POI boundary along with rigorous definitions.
- Given a POI name, we formulate our optimization problem in terms of maximizing $r^{\alpha}\mathcal{F}(c,r)$ under the quality constraint, where $\alpha \geq 0$ is the radius exponent.
- We introduce the optimal GeoSocialBound algorithm with linear scaling runtime complexity.
- We empirically evaluate the estimation performance of our algorithm for various radius exponents and POIs, and validate the complexity analysis.

2. PREVIOUS WORK

There have been two types of studies in the literature to reveal and utilize the characteristics of POIs: POI recommendation [1–3] and AOI discovery [4–6]. First, time-aware POI recommendations [1, 2] were introduced by observing that users tend to visit different places at different times from historical check-in records. A location-aware recommendation system was also proposed in [3], where an item-based collaborative filtering was employed to make POI recommendations. On the other hand, when people visit an AOI, they tend to check in and take photos. Therefore, it is of fundamental importance to find AOIs [4–6]. Previous studies on discovering AOIs were conducted mostly by using density-based clustering methods along with the collection of geo-tagged photos from LBSNs. Density-based spatial clustering of application with noise (DBSCAN) [5,6] is the most commonly used density-based clustering algorithm even if it was not originally designed for AOI discovery. DBSCAN can find multiple clusters with an overall average runtime complexity of $\mathcal{O}(n \log n)$ (the worst case complexity of $\mathcal{O}(n^2)$), where n denotes the total number of input records. Instead of using geo-tagged photos, geo-tagged textual data were utilized to discover an AOI, where correlations between textual descriptions (i.e., POI names) and locations were exploited in [8] and another clustering method based on the quality of regions for given POI names was developed in [9]. A heuristic for estimating a social POI boundary was also introduced in [10], where a two-phase algorithm with linear scaling complexity in the number of input records was performed. Moreover, the relationship between geo-referenced tweets of each nearby POI feature class and their location was investigated in [11] by means of manual, supervised, and unsupervised classification methods. In [12], deriving a boundary of imprecise regions was addressed by performing web page searches.

3. DATA ACQUISITION

We describe how to collect POIs and Twitter metadata associated with their locations.

3.1 Collecting POIs

To obtain a set of POIs and their centers, we use an open source database Geonames,[1] which has a great amount of geographical information and concepts. There are various geographical feature classes in the database such as spots, buildings, farms, roads, hills, rocks, etc., but the following four categorized types are taken into account for simplicity: observation point (S.OBPT), building (S.BLDG), stadium (S.STDM), and museum (S.MUS), which are summarized in Table 1.

Table 1: The POI types according to Geonames

POI type	**Geonames category**
Observation point	S.OBPT
Building	S.BLDG
Stadium	S.STDM
Museum	S.MUS

3.2 Collecting Twitter Data

We use a dataset collected via Twitter Streaming API. The dataset consists of a huge amount of geo-tagged tweets recorded from Twitter users from July 29, 2015 to August 29, 2015 (about one month) in the following two countries: the US and the UK. We removed the content that was automatically created by other services such as Tweetbot, TweetDeck, Twimight, and so forth. We see that each tweet contains a number of entities that are distinguished by their attributed field names. For data analysis, we adopt the following three essential fields from the metadata of tweets:

- *text*: actual UTF-8 text of the status update containing a POI name
- *lat*: latitude of the tweet's location
- *lon*: longitude of the tweet's location

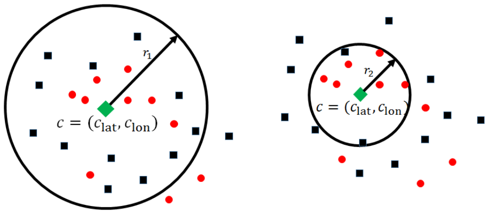
Four POIs placed in the US and the UK are used for our analysis, where a POI is selected for each Geonames category. Representative attributes of the four POIs are summarized in Table 2. The third column of Table 2 represents the POI center's coordinate $c = (c_{\text{lat}}, c_{\text{lon}})$, where c_{lat} and c_{lon} denote the latitude and longitude measured in degrees, respectively.

Table 2: Attributes of four POIs

POI name	**Geonames category**	**POI center** $c = (c_{\text{lat}}, c_{\text{lon}})$
London Eye	S.OBPT	51.503300^o, -0.119700^o
The White House	S.BLDG	38.8977^o, -77.0365^o
Dodger Stadium	S.STDM	34.073611^o, -118.24^o
Metropolitan Museum of Art	S.MUS	40.77891^o, -73.96367^o

Since users are able to tag a POI name (e.g., *#londoneye*) or insert it (e.g., *London Eye*) in their tweets to describe their interest in a POI, one can easily query all important records such as tweets containing a POI name. Through query processing, we obtain the filtered geo-tags whose text field is associated with the POI name. Since users tend to type the real-world terms of each POI into the tweet box, a POI name may be misspelled or have other words tacked on to it. We perform a keyword-based search by querying

[1] http://www.geonames.org


r_1
$c = (c_{\text{lat}}, c_{\text{lon}})$
r_2
$c = (c_{\text{lat}}, c_{\text{lon}})$

(a) Circle 1 (b) Circle 2

Figure 1: Sample datasets

semantically coherent but different words for a POI. Relevant variations for an annotation (i.e., a POI name) would include the hashtag followed by the POI name, its abbreviated name, its synonyms (if any), etc. (refer to Table 3). Then, the dataset can be partitioned into two subsets of geo-tagged tweets with and without the annotated POI names.

Table 3: Search queries

POI name	Search queries
London Eye	*#londoneye, London Eye, Londoneye, Millennium Wheel*
The White House	*#whitehouse, White House, Whitehouse*
Dodger Stadium	*#dodgerstadium, Dodger Stadium, Dodgerstadium*
Metropolitan Museum of Art	*#themet, #metropolitainmuseumofart, Metropolitan Museum of Art, The Met, Themet, Metropolitainmuseumofart*

4. PROBLEM FORMULATION

In this section, we formulate our optimization problem after formally defining some important terms used for our analysis.

4.1 Definitions

We start by introducing three important definitions in analyzing the estimation performance of social POI boundaries. The relevance of the data to a POI varies according to the geographic distance between the POI center and the locations where the data are generated. The tweets posted in locations far away from the POI center are expected to have no textual description for the POI. Hence, it is of utmost importance to formally define a social POI boundary and to estimate it with high accuracy. Unlike the DBSCAN algorithm that finds *arbitrarily-shaped* multiple clusters, to characterize a social POI boundary, we focus only on finding the maximum possible distance reachable from the POI center, thereby enabling to significantly reduce the computational complexity compared to other clustering methods.

Definition 1: Given a circle (c, r) that has the POI center $c = (c_{\text{lat}}, c_{\text{lon}})$ and a certain radius $r > 0$, let $\mathcal{D}(c, r)$ denote the set of all geo-tagged tweets that include a given textual description (i.e., a POI name) within the circle (c, r). Then, a social POI boundary is defined as a *convex hull type cluster* containing the given points in $\mathcal{D}(c, r)$.

Contrary to LBSNs with check-in records and geo-tagged photos, geo-tagged tweets with (semantically) different textual descriptions from a POI name may occur especially on Twitter. Such unimportant (or marginal) geo-tags may be acceptable since one cannot expect that all geo-tags in a given region contain the same textual description. For the sake of brevity, we denote geo-tags whose text contains and does not contain an associated POI name by "important" and "marginal" records, respectively. Similarly as in [9], to quantitatively measure the number of important records, we introduce the quality of an annotation in a given area.

Definition 2: Given a textual description (i.e., a POI name), the quality of a circle (c, r) is defined as

$$\begin{aligned}\text{Precision}(c, r) &= \frac{|\mathcal{D}(c, r)|}{|\mathcal{D}_{\texttt{all}}(c, r)|} \\ &= \frac{TP(c, r)}{TP(c, r) + FP(c, r)}, \qquad (1)\end{aligned}$$

which is the ratio of true positives to all predicted positives, where c and r denote the POI center and the radius of the circle. Here, $\mathcal{D}_{\texttt{all}}(c, r)$ denotes the set of all geo-tagged tweets within the circle (c, r); $TP(c, r) = |\mathcal{D}(c, r)|$ is the number of important records inside the circle (c, r); and $FP(c, r) = |\mathcal{D}_{\texttt{all}}(c, r) \setminus \mathcal{D}(c, r)|$ is the number of marginal records in the circle (c, r).

Example 1: Consider two circles having radii r_1 and r_2 in Figure 1, where $r_1 > r_2$. When red circles and black squares indicate important and marginal records within the circle (c, r_i), corresponding to $TP(c, r_i)$ and $FP(c, r_i)$, respectively, for $i \in \{1, 2\}$. The quality of each circle is computed by

Circle 1: $\text{Precision}(c, r_1) = \frac{8}{20} = 0.40$;

Circle 2: $\text{Precision}(c, r_2) = \frac{6}{8} = 0.75$.

Obviously, Circle 1 has a worse quality since it suffers from an overexpansion with a number of accompanying marginal geo-tags near the boundary of the circle. In this case, to reduce the number of marginal geo-tags, we need to decrease the radius of Circle 1, leading to an increment of the quality.

To measure estimation accuracy, we use the $\mathcal{F}$-measure, which is a popular measure of a test's accuracy for binary classification and thus can be considered as the most important score to judge a boundary. In our problem, we expect to divide the plane into two regions (i.e., an interior and an exterior) by finding a sufficiently large circle containing a large number of important records and a small number of marginal records.

Definition 3: Given a circle (c, r), the $\mathcal{F}$-measure is

$$\mathcal{F}(c, r) = \frac{2\text{Precision}(c, r)\text{Recall}(c, r)}{\text{Precision}(c, r) + \text{Recall}(c, r)}, \qquad (2)$$

which indicates the harmonic mean of $\text{Precision}(c, r)$ and $\text{Recall}(c, r)$. Here, $\text{Recall}(c, r)$ is the ratio of true positives to all actual positives, that is $\frac{TP(c,r)}{TP(c,r)+FN(c,r)}$; and $FN(c, r)$ is the number of important records outside the circle (c, r).

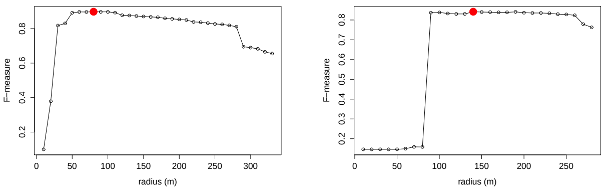
It is shown that the value of $TP(c, r) + FN(c, r)$ is fixed for a given dataset. Thus, $\mathcal{F}(c, r)$ depends on $TP(c, r)$ and $FP(c, r)$. More precisely, the $\mathcal{F}$-measure increases with high $TP(c, r)$ and low $FP(c, r)$.

Example 2: For the two circles in Figure 1, the $\mathcal{F}$-measure of each circle is computed by

Circle 1: $\mathcal{F}(c, r_1) = \frac{2 \times 0.4 \times 0.8}{0.4 + 0.8} = 0.53$;

Circle 2: $\mathcal{F}(c, r_2) = \frac{2 \times 0.75 \times 0.6}{0.75 + 0.6} = 0.67$.

From Example 2, one can see that $\mathcal{F}(c, r)$ becomes higher when there are relatively more important records and less marginal records inside the circle (c, r).



(a) London Eye (b) The White House

Figure 2: The $\mathcal{F}$-measure according to the radius

4.2 Optimization Problem

The $\mathcal{F}$-measure $\mathcal{F}(c, r)$ is evaluated according to the radius r, where London Eye and the White House as two POIs are considered in Figure 2a and 2b, respectively. The maximum $\mathcal{F}(c, r)$ is indicated by a red point. It is observed that $\mathcal{F}$-measure tends to be hardly degraded up to a certain critical point with respect to the radius r from the POI center. For this reason, although it is desirable to provide a social POI boundary with the highest $\mathcal{F}$-measure, it would also be good to considerably extend the social POI boundary at the cost of a *slightly reduced value of* $\mathcal{F}(c, r)$. For example, in Figure 2b, when the radius increases from 140m to 260m, $\mathcal{F}(c, r)$ decreases from 0.842 to 0.824, thus leading to only 2.14% reduction in $\mathcal{F}(c, r)$. Therefore, we formulate a new constrained optimization problem, in which we aim at finding the optimal radius such that the objective function expressed as the product of a power law in r (in meters) and an $\mathcal{F}$-measure $\mathcal{F}(c, r)$ is maximized, as follows:

$$r^* = \arg\max_{r \in (0, \bar{r}]} r^\alpha \mathcal{F}(c, r) \tag{3a}$$

$$\text{subject to} \quad \text{Precision}(c, r) \geq \eta, \tag{3b}$$

where Precision(c, r) and $\eta \in (0, 1)$ denote the quality of a circle (c, r) in Equation 1 and the target quality threshold (which will be specified in Section 6), respectively. Here, α is the radius exponent, balancing between different levels of geographic coverage, and $\bar{r}$ is an arbitrarily large radius such that Precision$(c, \bar{r})$ approaches almost zero.

5. GEOSOCIALBOUND: ESTIMATION ALGORITHM

The distribution of geo-tagged tweets is generally not homogeneous and significantly differs from each other according to POIs. In particular, the following interesting and insightful observations are made (refer to Figure 3): geo-tags are non-uniformly distributed over the geographic area; and there exist several important records *superimposed* at one point near the POI center. For these reasons, the objective function $r^\alpha \mathcal{F}(c, r)$ in Equation 3a is not necessarily concave, and thus it is hardly possible to find a closed-form solution.

In this section, we introduce a low-complexity estimation algorithm, namely GeoSocialBound, that solves the constrained optimization problem in Section 4.2. The overall procedure is summarized in Algorithm 1, which consists of two phases.

- **Phase 1** (Lines 2–10 in Algorithm 1): we find the maximum radius r_M such that Precision$(c, r) \geq \eta$ in Equation 3b.
- **Phase 2** (Lines 12–19 in Algorithm 1): we find the optimal radius r^* that maximizes $r^\alpha \mathcal{F}(c, r)$ in Equation 3a under the constraint in Equation 3b.

Algorithm 1: GeoSocialBound algorithm

```
   Input: $\mathcal{D}(c,\bar{r}), \mathcal{D}_{\texttt{all}}(c,\bar{r}), c, \Delta r, \alpha, \eta$
   Output: $\mathcal{D}(c, r^*)$
1  Initialization: $i \leftarrow 1; r_i \leftarrow \Delta r; r_M \leftarrow 0; r^* \leftarrow 0;$
     $N \leftarrow \frac{\bar{r}}{\Delta r}; TP(c,r_i) \leftarrow 0; FP(c,r_i) \leftarrow 0;$
     $FN(c,r_i) \leftarrow 0; \mathcal{F}(c,r_i) \leftarrow 0; \mathcal{D}_{\texttt{all}}(c,r_i) \leftarrow \emptyset;$
     $\mathcal{D}(c,r_i) \leftarrow \emptyset; \mathcal{F}(c,r^*) \leftarrow 0;$
     $\text{Precision}(c,r_i) \leftarrow 0; \text{Recall}(c,r_i) \leftarrow 0$
2  for $i \leftarrow 1$ to $N$ do
3      $\mathcal{D}_{\texttt{all}}(c,r_i) \leftarrow \text{Filter}(r_i, \mathcal{D}_{\texttt{all}}(c,\bar{r}))$ (see Algorithm 2)
4      $\mathcal{D}(c,r_i) \leftarrow \text{Filter}(r_i), \mathcal{D}(c,\bar{r}))$   (see Algorithm 2)
5      $TP(c,r_i) \leftarrow |\mathcal{D}(c,r_i)|$
6      $FP(c,r_i) \leftarrow |\mathcal{D}_{\texttt{all}}(c,r_i) \setminus \mathcal{D}(c,r_i)|$
7      $\text{Precision}(c,r_i) \leftarrow \frac{TP(c,r_i)}{TP(c,r_i)+FP(c,r_i)}$
8      if $r_i > r_M$ and $\text{Precision}(c,r_i) \geq \eta$ then
9          $r_M \leftarrow r_i$
10     $r_{i+1} \leftarrow r_i + \Delta r$
11 $M \leftarrow \frac{r_M}{\Delta r}$
12 for $i \leftarrow 1$ to $M$ do
13     $FN(c,r_i) \leftarrow |\mathcal{D}(c,r_M) \setminus \mathcal{D}(c,r_i)|$
14     $\text{Recall}(c,r_i) \leftarrow \frac{TP(c,r_i)}{TP(c,r_i)+FN(c,r_i)}$
15     $\mathcal{F}(c,r_i) \leftarrow \frac{2\text{Precision}(c,r_i)\text{Recall}(c,r_i)}{\text{Precision}(c,r_i)+\text{Recall}(c,r_i)}$
16     if ${r_i}^\alpha \mathcal{F}(c,r_i) \geq {r^*}^\alpha \mathcal{F}(c,r^*)$ then
17         $r^* \leftarrow r_i$
18         $\mathcal{F}(c,r^*) \leftarrow \mathcal{F}(c,r_i)$
19     $r_{i+1} \leftarrow r_i + \Delta r$
20 return $\mathcal{D}(c,r^*)$
```

Algorithm 2: Filter$(r_i, \mathcal{G}(c, \bar{r}))$

```
   Input: $\mathcal{G}(c,\bar{r})$ ($\mathcal{D}(c,\bar{r})$ or $\mathcal{D}_{\texttt{all}}(c,\bar{r})$), $c, r_i$
   Output: $\mathcal{G}(c,r_i)$ for $r_i \in (0,\bar{r}]$
1  Initialization: $k \leftarrow 1; l \leftarrow |\mathcal{G}(c,\bar{r})|$
2  for $k \leftarrow 1$ to $l$ do
3      $\mathcal{G}(c,r_i) \leftarrow \{q_k | d(c,q_k) \leq r_i\}$
4  return $\mathcal{G}(c,r_i)$
```

In the algorithm, $\bar{r}$ is assumed to be arbitrarily large fulfilling Precision$(c, r) \simeq 0$. We suppose that two classified sets $\mathcal{D}(c, \bar{r})$ and $\mathcal{D}_{\texttt{all}}(c, \bar{r})$ are given as inputs. Since the geo-tags' coordinate is expressed in finite precision, we perform uniform sampling from $(0, \bar{r}]$ with a given sampling interval $\Delta r > 0$, where Δr is set to a small value greater than the GPS error distance. In each phase, given a POI, we start by using a circle centered at c with radius $r_i = \Delta r$. The radius is increased by Δr for each step, and r_M in Phase 1 or r^* in Phase 2 is updated iteratively if a certain condition is fulfilled. For each iterative step in Phase 1, the subroutine Filter$(r_i, \mathcal{G}(c, \bar{r}))$ in Algorithm 2 is invoked to collect the set $\mathcal{G}(c, r_i) \subseteq \mathcal{G}(c, \bar{r})$, where $\mathcal{G}(c, r_i)$ is either $\mathcal{D}(c, r_i)$ or $\mathcal{D}_{\texttt{all}}(c, r_i)$. Then, it follows that $\mathcal{G}(c, r_i) = \{q_i | d(c, q_i) \leq r_i\}$, where $d(c, q_k)$ denotes the geographic distance between c and the k^{th} geo-tag's coordinate q_k in $\mathcal{G}(c, \bar{r})$. Line 3 in Algorithm 2 can be implemented by using a searching algorithm such as *R-trees*.

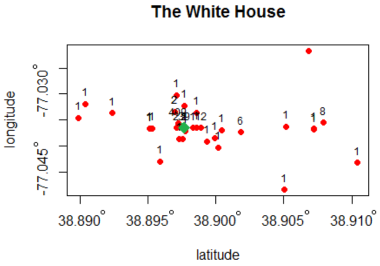


Figure 3: Superimposed important records (red) near the White House (green), where the latitude and longitude are measured in degrees and the number of superimposed tags is marked at each point

Thereafter, by solving the convex-hull problem (e.g., *quickhull*), we can finally obtain the smallest convex polygon that contains all the geo-tags in the set $\mathcal{D}(c, r^*)$ quickly, corresponding to the estimated social POI boundary.

6. ANALYSIS RESULTS AND DISCUSSION

In this section, using the proposed GeoSocialBound algorithm in Section 5, we first analyze the overall average computational complexity and then show experimental results. We simply assume $\eta = 0.5$, which can also be set to another value to control the quality constraint, and $\Delta r = 10m$ for all POIs. Even if $\bar{r}$ is assumed to be arbitrarily large, we need to set $\bar{r}$ to a reasonable value in our experiment so as to efficiently reduce the complexity of the algorithm. In our work, we assume $\bar{r} = \gamma r_{\texttt{cover}}$, where γ is a positive constant and $r_{\texttt{cover}}$ denotes an *approximate* radius from the POI center covering the geographic area of each POI and is obtained from Google Maps Geocoding API.[2] The Google Maps Geocoding API returns the bounding box of a POI center, constructed by the southwest and northeast corners, and the value of $r_{\texttt{cover}}$ is given by computing the maximum of the two distances between the southwest corner and the POI center and between the northeast corner and the POI center. While γ can be properly determined according to the POI types (e.g., $\gamma = 1$ for a cafeteria and $\gamma = 10$ for an observation point), we use $\gamma = 10$ for all POIs.

6.1 Computational Complexity

We analyze the runtime complexity of the proposed GeoSocialBound algorithm. Let us denote the cardinality of the set $\mathcal{D}_{\texttt{all}}(c, \bar{r})$ by $n_{\texttt{all}} \triangleq |\mathcal{D}_{\texttt{all}}(c, \bar{r})|$. In our method, the overall complexity is dominated by Algorithm 2, which is $\mathcal{O}(n_{\texttt{all}})$; therefore, the overall runtime complexity is given by $\mathcal{O}(n_{\texttt{all}})$. In Figure 4, the overall runtime versus $n_{\texttt{all}}$ is evaluated for $\alpha = 1$, where the social POI boundary of the White House is estimated using the dataset in Section 3. An asymptotic line with a proper bias is also shown in Figure 4, showing trends consistent with our experimental result.

6.2 Experimental Results

Using the GeoSocialBound algorithm in Section 5, our experimental results are shown according to different values of $\alpha \geq 0$. Statistics of four POIs in Table 2, including $|\mathcal{D}(c, \bar{r})|$ (i.e., the number of important records), $|\mathcal{D}_{\texttt{all}}(c, \bar{r})|$, and $r_{\texttt{cover}}$, are shown in Table 4. As depicted in Table 4, since four POIs were selected according to each different Geonames category, both $|\mathcal{D}(c, \bar{r})|$ and $|\mathcal{D}_{\texttt{all}}(c, \bar{r})|$ are significantly different from each other. The estimation performance of the four POIs is summarized in Table 5, where $\alpha \in \{0, 1\}$. From the table, our interesting finding is that when the radius exponent α varies from 0 or 1, $\mathcal{F}(c, r^*)$ is slightly reduced, but the optimal radius r^* is remarkably increased, thus providing an expanded geographic coverage at the cost of at most 9.59% reduction in $\mathcal{F}(c, r^*)$. In addition, the optimal radius r^* versus the radius exponent α is evaluated in Figure 5, where $\alpha \in [0, 1]$ and four POIs are considered. From the figure, we observe that r^* is monotonically non-decreasing with α and is hardly increased beyond a certain value of α near 0.2. This reveals that changing the value of α may have little impact on the estimation performance under the regime $\alpha \in [0.2, 1]$. To better understand our technique, the estimated social POI boundaries are illustrated in Figure 6, where the green pin is the POI center and the red pins are the important records.

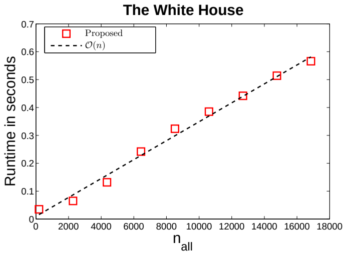


Figure 4: Runtime complexity

Table 4: Statistics of the four POIs

POI name	$\lvert\mathcal{D}(c, \bar{r})\rvert$	$\lvert\mathcal{D}_{\texttt{all}}(c, \bar{r})\rvert$	$r_{\texttt{cover}}$ (m)
London Eye	1,397	17,547	176
The White House	510	17,644	188
Dodger Stadium	828	17,875	150
Metropolitan Museum of Art	507	19,396	239

Table 5: The estimation performance

POI name	α	r_M (m)	r^* (m)	$\mathcal{F}(c, r^*)$	$r^{*\alpha}\mathcal{F}(c, r^*)$
London Eye	0 1	330	80 280	0.897 0.811	0.897 227.08
The White House	0 1	280	140 260	0.842 0.824	0.842 214.24
Dodger stadium	0 1	1,500	660 1,500	0.951 0.903	0.951 1,354.5
Metropolitan Museum of Art	0 1	170	150 170	0.760 0.757	0.706 128.69

7. CONCLUSION AND FUTURE WORK

In this paper, we first characterized a social POI boundary utilizing spatio–textual information on Twitter. Specifically, we introduced a constrained optimization problem formulation in terms of maximizing $r^\alpha \mathcal{F}(c, r)$, where α is the radius exponent, and an efficient optimal estimation algorithm with

[2] https://developers.google.com

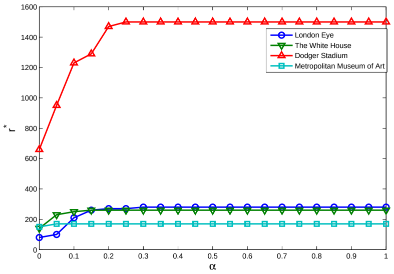
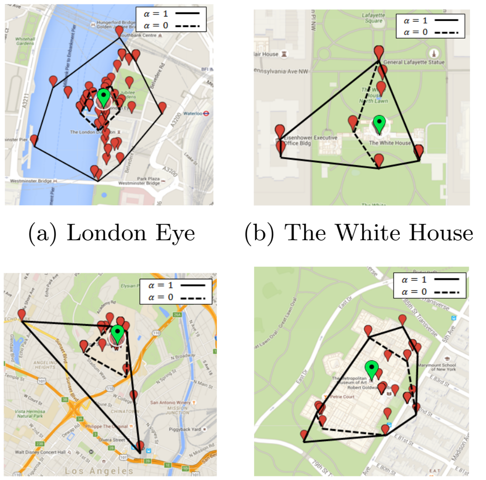


Figure 5: The optimal r^* according to the radius exponent α



(a) London Eye (b) The White House

(c) Dodger Stadium (d) Metropolitan Museum of Art

Figure 6: Estimated social POI boundaries

linear runtime complexity. In addition, we numerically evaluated the estimation performance for various α's and POIs while delineating the estimated social POI boundaries.

To further improve the estimation performance, future work in this area includes a study on a joint optimization of the radius and the POI center by allowing to update the POI center.

8. ACKNOWLEDGMENT

D. D. Vu and W.-Y. Shin's work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2014R1A1A2054577) and by the Ministry of Science, ICT & Future Planning (MSIP) (2015R1A2A1A15054-248). H. To and C. Shahabi's work was funded by NSF grants IIS-1320149, CNS-1461963, the USC Integrated Media Systems Center, and unrestricted cash gift from Northrop Grumman. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors.

9. REFERENCES

[1] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*, pages 363–372, July 2013.

[2] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*, pages 325–334, July 2011.

[3] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. LARS: A location-aware recommender system. In *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE2012)*, pages 450–461, April 2012.

[4] J. Liu, Z. Huang, L. Chen, H.-T. Shen, and Z. Yan. Discovering areas of interest with geo-tagged images and check-ins. In *Proceeding of the 20th ACM International Conference on Multimedia (MM'12)*, pages 589–598, October 2012.

[5] S. Kisilevich, F. Mansmann, and D. Keim. P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st International Conference and Exhibition of Computing for Geospatial Research & Application (COM.Geo2010)*, June 2010.

[6] M. Ester, H.-P. Keriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Data Mining and Knowledge Discovery*, 96(34): 226–231, 1996.

[7] W.-Y. Shin, B. C. Singh, J. Cho, and A. M. Everett. A new understanding of friendships in space: Complex networks meet Twitter. *Journal of Information Science*, 41(6): 751–764, 2015.

[8] S. V. Canneyt, S. Schockaert, O. V. Laere, and B. Dhoedt. Detecting places of interest using social media. In *Proceedings of the the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT'12)*, pages 447–451, December 2012.

[9] A. Skovsgaard, D. Sidlauskas, and C. S. Jensen. A clustering approach to the discovery of points of interest from geo–tagged microblog posts. In *Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management (MDM'14)*, pages 178–188, July 2014.

[10] D. D. Vu and W.-Y. Shin. Low-complexity of POI boundaries using geo-tagged tweets: A geographic proximity based approach. In *Proceedings of the 8th ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN'15)*, November 2015.

[11] S. Hahmann, R. S. Purves, and D. Burghardt. Twitter location (sometimes) matters: Exploring the relationship between georeferenced tweet content and nearby feature class. *Journal of Spatial Information Science*, 9: 1–36, 2014.

[12] A. Arampatzis, M. van Kreveld, I. Reinbacher, C. B. Jones, S. Vaid, P. Clough, H. Joho, and M. Sanderson. Web-based delineation of imprecise regions. *Computers, Environment and Urban System*, 30(4): 436–459, 2006.