Automatically Recognizing Places of Interest from Unreliable GPS Data Using Spatio-temporal Density Estimation and Line Intersections

Tanusri Bhattacharya[a,*], Lars Kulik[a], James Bailey[a]

[a]*Department of Computing and Information Systems, The University of Melbourne, Parkville, Australia*

Abstract

Stay points are important for recognizing significant places from a mobile user's GPS trajectory. Such places are often located indoors and in urban canyons, where GPS is unreliable. Consequently, mapping a user's stay point to a Place of Interest (POI) using only GPS data is particularly challenging. Our novel algorithm employs both spatio-temporal density estimation and line count inference to predict and rank a user's POI(s) at building level accuracy from noisy time-annotated GPS data points. An experimental study demonstrates the superiority of our algorithm against several baseline approaches with a recall of 96.5% for the top 5 retrieved locations.

Keywords: Indoor GPS, Urban canyons, Trajectory Mining, Significant Places

1. Introduction

The rise of GPS-enabled mobile devices has created exciting opportunities for tracking people, objects and vehicles, as well as animals using their digital traces. Such traces or trajectories can be mined to derive knowledge about the behaviour of moving objects for a range of purposes: for health monitoring [1, 2], offender monitoring [3], animal migration [4–6], traffic and transportation management systems [7–9], location-based services and social networks [10–13]. For example, in case of traffic management systems location traces can be mined to estimate traffic delays on the road [7, 8]. Rich monitoring of traffic and road conditions such as bumps and breaks, can also be performed with the help of additional sensors along with GPS [9].

Although satellite-based GPS positioning provides good coverage for modern navigation and tracking; it does not work well for indoor places or urban canyons, due to poor line-of-sight transmission between the receiver and the satellites. For a GPS device to be accurate, a clear line of sight to satellites is necessary, so that signals from at least four satellites are available to the GPS receiver for computing its current position [14].

The complexity of indoor environments, which contain walls, equipment, people and other objects which can influence electro-magnetic waves, causes the multipath propagation of a GPS signal and severely reduces the accuracy of positioning. In urban canyons, the multipath effect and signal loss are also caused by surrounding tall buildings [15]. The High Sensitivity GPS (HSGPS) receiver has been designed to allow tracking in such critical environments. It is claimed a HSGPS receiver is able to acquire signals down to -190 dBW to allow tracking in indoor locations and urban areas [15, 16]. Unfortunately, today's GPS-enabled mobile phones (even those that can use A-GPS) still suffer from poor accuracy when being used in indoor environments or urban canyons. The median accuracy of positioning in such environments is as high as 70 meters [17]. Kjrgaard et al. [15] have investigated the extent to which GPS can be used for positioning in indoor places. Their study showed that if using a state-of-the art GPS receiver (High-Sensitivity-GPS), the accuracy of GPS positioning is adequate (within 10 meters) in many wooden house and brick/concrete buildings. However, the availability and performance of positioning is much lower when using the embedded GPS-receiver in current mobile phones. They showed the performance of GPS positioning for an indoor place not only

*Corresponding author

Email addresses: tanusrib@student.unimelb.edu.au (Tanusri Bhattacharya), lkulik@unimelb.edu.au (Lars Kulik), baileyj@unimelb.edu.au (James Bailey)

depends on the type of receiver, but also on nearby building elements and materials, the number of walls, the number of floors and the composition of surrounding buildings.

A range of prior approaches have proposed the use of other technologies such as RFID, Wi-Fi or Bluetooth to improve location accuracy [18–26] in GPS signal degraded areas. The commercial products like Apple's Maps or Google Maps installed in modern smartphones also mostly rely on Wi-Fi or mobile network connections to perform localization at places where GPS signal is unavailable. The opportunity to use such technologies is dependent on the building-infrastructure and will not be always available. Typically, there is also additional cost for implementation. Moreover, these localization techniques are not designed for emergency critical scenarios such as fire-fighting and earthquake or tsunami response. Other studies have proposed PDR (Pedestrian Dead Reckoning) based techniques for use in indoor locations [27]. D'Souza et al. has proposed techniques for real-time user tracking in indoor environments using wearable motion sensors within a floor-plan and limited wireless sensor network [28]. However these techniques are specialized for indoor environments and are not designed for outdoor use.

Indoor places play an important role in daily life and people spend a large amount of their time in indoor locations such as homes, offices, universities, or restaurants [29]. GPS provides broad coverage for positioning worldwide and provides excellent accuracy for outdoor environments, without incurring any additional infrastructure cost for indoor positioning. The pervasiveness of GPS enabled mobile devices is promoting the generation of increasing volumes of trajectories reflecting people's daily movement behaviour. Prior literature has examined how to extract segments of a GPS trajectory corresponding to user's stay at different significant places, such as the home, office, park or shopping mall [12, 30–36]. A so-called *Stay Point* can be computed as the mean of the extracted GPS points. However, significant places are often indoor locations and may correspond to buildings in urban areas where GPS is less accurate. Therefore obtaining the user's Place of Interest (POI) by mapping the stay point to a POI database is not feasible. For example, Figure 1 shows the measured GPS positions by a mobile device for a user's stay in building A at The University of Melbourne. The centroid or mean of the GPS points is shown in orange but the nearest building to the so called *stay point* does not correspond to the true POI. The problem becomes even more challenging when a user stays at multiple nearby buildings in urban canyons, while being at a significant place. For example, a user's office can be located within a building in a city-centre, but she may have lunch in a nearby restaurant. In a university, a user may spend time in a library and can attend a seminar in another nearby building. In these scenarios, estimating the user's true POIs at the building level becomes impossible using the *Stay Point* technique, since the measured GPS positions often return a single cluster of GPS points due to poor accuracy.

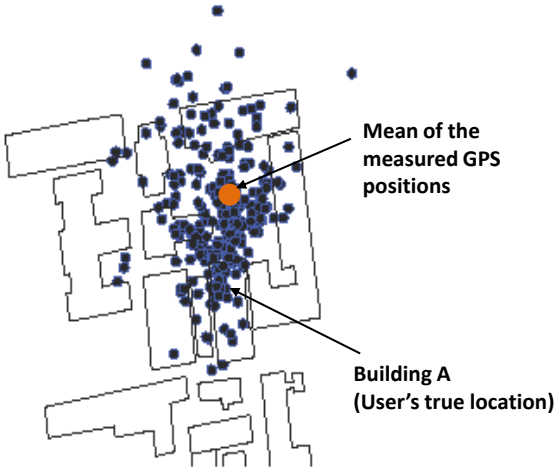

Figure 1: Measured GPS positions by a mobile device for a user's stay at a university building A. The mean of the GPS positions (shown in orange), known as *stay point*, is far away from the user's true POI (building A).

To meet these challenges, we propose a novel algorithm (*POI-ID*) to obtain a user's POI(s) corresponding to his/her stay at a significant place from a set of highly inaccurate GPS data points. Given a set of time-stamped GPS points and a list of surrounding POIs, *POI-ID* produces a ranking of the POIs according to the user's most likely location at a building level accuracy.

Applications like emergency search and rescue operations require a user to be tracked at room level. However, predicting places accurately at the building level using only GPS data has many applications as well. It can help to improve the context-awareness of GPS-enabled mobile devices by understanding the semantic meanings of locations. A number of mobile social-networking applications such as Foursquare [11], Google+ [37] and Facebook Places [38] allow users to *check-in* to a location, for example upon arrival at a restaurant or cinema. Other applications, such as Instagram [39], allow users to select a location from a list, when sharing a photo. User input on a mobile device is typically considered to be time consuming and predicting likely choices is a key technique in minimizing user input. Predicting a user's true POI(s) from a given set of coordinates and showing them within the top few positions of the displayed list, is highly desirable to improve the user's experience in such applications. Moreover, accurate prediction of user's true POI at the first position in the list can assist with automatic check-ins to a place. Our algorithm (*POI-ID*) can be very useful in these applications for estimating the user's POI(s) with high accuracy. We will see later, that *POI-ID* can retrieve a user's true POI(s) within the top 5 results for 96.5% of searches and can identify the exact POI at position one for 56% of searches, without using any prior knowledge about the user (such as location history) or the place (such as popularity). Our proposed algorithm may also be applicable in health care services to monitor places being visited by ill or elderly people and alert them accordingly about potential hazards. *POI-ID* might also be extended for operation in real-time offender-monitoring applications, to estimate the places visited by the criminals, even when such places are not equipped with Wi-Fi, RFID or other localization technologies.

A key contribution is the consideration of the spatial uncertainty of each measured GPS point for obtaining the most likely POI. According to the USA's National Standard for Spatial Data Accuracy (NSSD), the horizontal error of GPS positioning is assumed to follow a circular normal distribution [40]. GPS devices report their position accuracy as the radius or diameter of the contour of the distribution as shown in Figure 2.

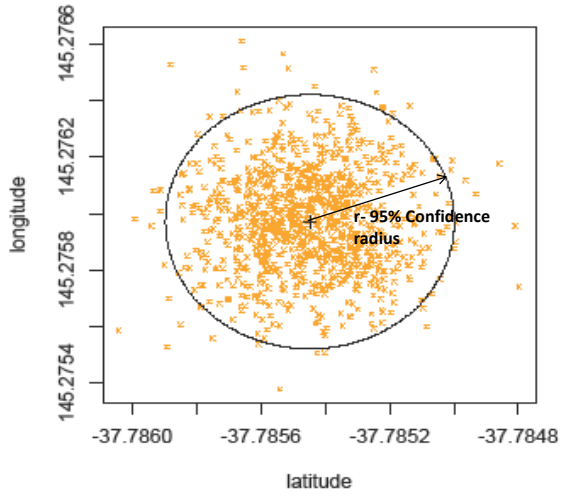

Figure 2: GPS horizontal position accuracy as 95% confidence circle with mean (centre of the circle) as the reported position

We propose a two phase algorithm. In the first phase, for each uncertain GPS point, we generate random points within the uncertainty circle according to the error distribution. We then perform kernel density estimation in three dimensions - latitude, longitude and time on all the generated random points. The POIs are then ranked according to the spatio-temporal density.

In the second phase, we rank the POIs using a line segment intersection based approach. Line segments are obtained by joining individual GPS points in order of their time stamps. We hypothesize that if the user is inside a POI, the number of GPS line segments intersecting that POI will be much higher than if the user is simply moving around it. Another key contribution we make is the development of theoretical models to support this argument, based on an extension of the classic Buffon-Laplace needle problem [41]. We compute the expected values of GPS line segments intersecting the POI in both *inside* and *moving around* scenarios. Each POI is ranked according to the number of intersecting line segments. The final ranking of the POIs is computed according to the Mean Reciprocal Rank (MRR) of the two rankings obtained in the above two phases.

Our main contributions can be summarized as follows:

- We develop an algorithm that uses inaccurate measurements from GPS devices and intersecting GPS line segment counts to predict and rank the most likely POI(s) corresponding to a user's stay at an indoor place.
- We propose theoretical models to compute the expected value of the number of GPS line segments intersecting a POI, assuming a user is *inside* or *moving around* it.
- We provide an experimental study that demonstrates that our algorithm is highly successful in predicting a user's actual POI, even for indoor places in crowded urban regions, and that it outperforms several baseline methods.
- We show that without using any other proximity or indoor localization technologies such as RFID, Wi-Fi or Bluetooth; or using any other prior knowledge such as user's location histories or popularity of places, *POI-ID* can estimate a user's most likely POI(s) at a building level of accuracy.

2. Related Work

The related literature can be broadly classified into two categories: 1) *place extraction algorithms* which extract segments from user's trajectories corresponding to significant places and 2) *place ranking algorithms* which retrieve and rank the nearby POIs according to a user's current location coordinate. As summarized in Table 1, previous works cannot identify a user's true POI(s) in an urban region only using GPS data and in the absence of prior knowledge about the user or the place.

2.1. Place Extraction or Learning Algorithms

The existing literature related to extracting or learning significant places can be categorized as: 1) geometry-based techniques and 2) fingerprinting-based techniques. The first category mostly uses GPS technology for location acquisition and often extracts places as the centres of individual clusters of geographical coordinates. However, knowing the user's true POI at building level accuracy, specifically in urban regions, was not possible with these techniques due to too much GPS noise at indoor locations. On the other hand, the fingerprinting-based techniques use cell towers and Wi-Fi access points to learn the places more accurately without providing any location coordinates. Our algorithm is similar to geometry-based techniques since we are also using GPS technology for location acquisition. However unlike the previous works, we have focused on estimating and ranking a user's true POI(s) at building level accuracy using only GPS data.

Geometry-based techniques: Many prior works have used clustering based techniques to extract clusters of GPS fixes representing significant places. However, most of these techniques cannot identify places if there is no GPS fix at all. Ashbrook et al [30], used a variation of k-means clustering to extract places. Kang et al. proposed a time-based clustering algorithm to cluster the GPS fixes corresponding to significant places [22]. Though the user does not have to provide the number of clusters or stop locations a-priori, this technique still depends on the silent parameters *distance* and *time* thresholds for obtaining the clusters. Moreover, the technique appears to be more effective on the location coordinates obtained from Wi-Fi access points rather than GPS fixes. Palma et al. [32] used a variation of DBSCAN [42] algorithm to identify low-speed segments of the trajectory as potential places. In [31], stops (places) and moves are computed by continuously checking the intersection of the trajectory with the geometry of a user-given set of POIs. For a long trajectory the computation cost of this technique can be high.

Other techniques have considered time and distance parameters to extract places from a GPS trajectory. In [43], Marmasse and Schmandt have extracted stay locations based on the fact that GPS signals get lost once the mobile user is inside a building. This technique is not able to identify outdoor places such as a park, bus-stop or open market place. Moreover, with the advancement of GPS receivers, today's smart-phones can obtain GPS fixes for many indoor places, depending on the building material and construction. Hariharan et al. have extracted *stays* by considering *roaming distance* and *time duration* [44]. Though their method can detect both indoor and outdoor places, the time complexity of the technique is very high. Zheng et al. [12, 33, 34] have extracted stay points from a GPS trajectory. A stay point is regarded as a virtual location characterized by consecutive GPS points which are within a predefined time and distance threshold. However, in practice, it can be very difficult to choose appropriate threshold values.

In [35], a stay point is identified through the temporal interval between consecutive GPS points. Their technique is only applicable for car trajectories and is not suitable for mobile users. In [36], Bhattacharya et al. identified segments of a trajectory as potentially significant places by considering the bearing change distribution of GPS fixes along with speed and acceleration. The low speed segments of the trajectory that have a higher variation in bearing change are considered as significant places. This technique can identify many significant places irrespective of their size and time duration. However, selection of the proper parameters still remains crucial for the success of the algorithm. Xiao et al. has addressed the issue of predicting a user's POI by expanding a stay-point to a stay region with a uncertainty parameter γ [45]. A feature vector is computed for each stay region by giving weight to each POI category in that region using a TF-IDF (Term Frequency-Inverse Document Frequency) methodology. However, unlike our algorithm, their technique involves the use of multiple user trajectories.

Nurmi et al. used a non-parametric probabilistic model, the Dirichlet process mixture model, to predict places (home and work locations) without considering the time dimension of data points [25]. Liao et al. have proposed a hierarchically structured Conditional Random Field (CRF) to predict a user's activity and significant places from GPS traces [46]. Recently Lee et al. [47], used a superstate model, an extension of the Hidden Markov Model (HMM), to extract places from GPS logs. Their model has been claimed to have better detection accuracy than many previous algorithms.

Fingerprinting-based techniques: Laasonen et al. [18] proposed to learn significant places based on the currently connected cell towers. They clustered the cell towers and identified the clusters as places with a time duration greater than a threshold value. In [19], Krumm et al. inferred the transition of states from "still" to "moving", by examining the variance of the strongest Wi-Fi access points and using an HMM based model. Given the fact that the Wi-Fi signal strength will vary with locations, they developed a HMM-based location model with locations as nodes and transition probabilities determined by the user's speed, current inferred state ("still" or "moving") and by the building's floor plans. In [21, 26], techniques were proposed to automatically learn places by continuously monitoring the radio environment to look for new *beacons* around a mobile device. The major challenge here is how to deal with infrequent beacons appearing during a mobile user's stay at a place. *BeaconPrint* [21] has used multiple scan windows to distinguish infrequently seen beacons from those seen when a mobile user leaves a place. However, this technique cannot detect places such as a market where the radio beacons are highly inconsistent, due to continuous movement of the user or high radio inference. A more robust technique, the *PlaceSense* algorithm, has been proposed in [26]. It uses separate mechanisms for detecting the entrance and exit to a place. *PlaceSense* recognizes the entrance to a place from the newly seen beacons, but avoids unstable beacons. The exit from a place is determined by the "disappearance of representative beacons seen in that place." [26]. Kim et al. used a combination of Wi-Fi, accelerometer and GPS data to predict a user's places, movements and paths, respectively [48]. In [49], Brouwers et al. performed a comparative study using three different sensor sources (GPS, Wi-Fi and Geolocation) to detect a mobile user's dwelling location in an urban canyon. For a GPS sensor, if an identified cluster of GPS points is within 100 meters of the user's ground-truth location, they considered it to be a True Positive (TP). Unlike their work, we consider a retrieved POI to be TP only when the user is truly located inside it.

2.2. Place Ranking Algorithms

Another collection of literature aims to map a user's geographic coordinates (latitude and longitude) to a POI. Knowing a user's POI from a given latitude longitude pair is particularly important in applications such as user *check-ins* in social network services like Foursquare. Substantial research has been carried out to retrieve and rank the nearby places given a user's location coordinates. Our research is different in that we are considering GPS fixes over a time period during a user's stay at a place, instead of a single GPS point at a time-stamp. Unlike these techniques, we don't consider *check-ins* from multiple users visiting a single place for ranking purposes (though our algorithm could certainly be extended to use such information). Moreover, our algorithm can be adapted for automatic check-in services, provided a user's locations are being tracked over some time period.

In [50], the authors proposed using the contextual and behavioural similarity between users, in addition to location and time to retrieve POIs. Lian et al. used a 545 user check-in history in a city to map a user's current location to a POI dataset consisting of 16,000 POIs. They used different features such as distance, time, users check-in history, POI popularity to train their model which can predict locations with 64.5% recall within the top 5 retrieved places [51]. Shaw et al. have considered some additional features, for example, the number of users currently checked-in the

place, user's personal history of visiting the place including the time of day and the number of user's friends currently checked-in to rank the POIs [17].

Table 1: Comparison of the literature related to place extraction or ranking algorithms

Literature	Location technology	Methodology	Trajectory/data type	Prior knowledge	POI detection in urban region (?)
[22, 30, 32]	GPS/Wi-Fi	Clustering	Single user	Not required	Not suitable
[35, 43]	GPS	Time threshold	Mobile user/Taxi trajectories	Not required	No suitable
[12, 33, 34, 44]	GPS	Time and distance threshold	Single/multiple user(s)	location history	Not suitable
[31]	GPS	Checking intersection with POI(s)	Single user	Not required	Not suitable
[45]	GPS	Time and distance threshold	Multiple users	POI categories	Unknown
[36]	GPS	Speed, acceleration and bearing change	Single user	Not required	Not suitable
[25, 46, 47]	GPS/RFID	Probabilistic model	Single user	Not required	Not suitable with GPS
[18]	Cell Tower	Cell tower clustering	Single user	Not required	Not suitable
[19]	Wi-Fi	Hidden Markov Model	NA	Not required	Yes
[21, 26]	Radio beacon	Monitoring surrounded radio environment	NA	Not required	Yes
[48]	Wi-Fi/GPS	Monitoring stable Wi-Fi signals for indoor place detection. GPS and accelerometer had been used for path and movement detection.	Single user	Not required	Yes. Only with Wi-Fi data
[49]	Wi-Fi/GPS/Geolocation	Supervised learning using GPS/Wi-Fi and Geolocation services.	Single user	Not required	Not suitable with GPS
[17, 50, 51]	GPS	Supervised machine learning techniques	Multiple users' location check-ins	Popularity of a place, check-in histories, behavioural similarity among users	Yes
Proposed algorithm (POI-ID)	GPS	Unsupervised learning using only GPS data	Single user	Not required	Yes, at building level accuracy

3. Estimating Places of Interest

In this section we will describe our methodology and algorithm used to predict a user's POI from highly noisy GPS data points. We will first define some required terminology.

3.1. Definitions

Horizontal GPS Accuracy. The horizontal accuracy, a, of a measured GPS position $l = (x, y)$ at a time-stamp t is defined as the radius of a confidence circle centering the position l assuming the positioning error to follow a circular normal distribution. This is described in Figure 2. The confidence circle thus indicates the probability of the true location to be inside that circle.

GPS Data Point. A GPS Data Point $p = (x, y, a, t)$ is a measured GPS position (x, y) with latitude x and longitude y at time-stamp t with a reported horizontal GPS accuracy a (as reported by the device).

GPS Trajectory. A GPS Trajectory T is a time-stamped consecutive sequence of GPS data points $p_i \in P, P = \{p_1, p_2, \ldots, p_n\}$ such that $\forall i \in [1, n], p_i = (x_i, y_i, a_i, t_i)$ and $t_i < t_{i+1}$, where x_i, y_i, a_i and t_i represent latitude, longitude, horizontal accuracy and time-stamp, respectively.

Place Data. Place data L is a consecutive time-stamped sequence of GPS data points corresponding to a user's stay at a place where $L \subseteq T$ and $L = \{p_{i+1}, p_{i+2}, \ldots, p_{i+m}\}$ such that each $p_{i+j} = (x_{i+j}, y_{i+j}, a_{i+j}, t_{i+j})$ and $t_{i+j} < t_{i+j+1}$ for $1 \leq j \leq m$.

POI Dataset. A POI Dataset is a set of polygonal POIs $POI = \{poi_1, poi_2, \ldots, poi_q\}$ such that $\forall j \in [1, q], poi_j = (typ_j, add_j, bdr_j)$ where typ_j and add_j are the type (for example, home, office, university, childcare, etc) and street address or building name of poi_j, respectively. The bdr_j represents the boundary of poi_j and is defined as $bdr_j = \{c_1, c_2, \ldots, c_k\}$ such that $\forall i \in [1, k], c_i$ is a (latitude,longitude) pair representing the individual corner location of the k-sided polygon.

3.2. Methodology

We perform an initial data preprocessing step based on the instantaneous speed of a mobile user. People either remain still or walk within a confined region in a significant place such as the home, office, university or shopping mall. Therefore we are interested only in the GPS data points where the instantaneous speed is not more than the maximum walking speed of the mobile user. The average human walking speed normally varies from 1.2 to 1.5 meter/sec [52, 53]. However, according to the previous studies [54–56], 2 meter/sec is the cut-off speed beyond which people normally prefer to run rather than to walk to save energy. Hence we chose 2 meter/sec as the maximum walking speed of a mobile user while being at a significant place. We remove all the GPS fixes as noise at which a user's speed is greater than 2 meter/sec. A similar preprocessing step is also performed while implementing the baseline techniques in Section 4. The speed at a GPS fix is obtained through i) a direct measurement of the GPS receiver or ii) computation based on the Euclidean distance between two consecutive GPS fixes and the time interval between them.

We propose a two-phase algorithm to estimate a user's most likely POI. In the first phase we rank the POIs according to the spatio-temporal closeness of uncertain GPS data points using trivariate Kernel Density estimation [57]. In the second phase we rank the POIs using a line segment intersection based approach.

3.3. First Phase: Ranking POIs by Kernel Density Estimation (Spatio-temporal Density Rank)

We first rank the POIs using three dimensional (latitude, longitude and time) Kernel Density estimation. Our contribution is the use of the horizontal positioning accuracy of each uncertain GPS data point as reported by the device. Figure 3 describes the confidence circles of the measured GPS positions corresponding to a user's stay at place A. We consider the fact that the measured GPS location is not the user's true location. There are different sources that contribute to error in the GPS measurement, for example, atmospheric effects, multipath signals, ephemeris, clock error and poor satellite geometry [14]. GPS position accuracy is mainly governed by the Dilution of Precision (DOP) [14]. The horizontal position accuracy is reported assuming the GPS fixes follow a bivariate normal distribution with no correlation between x and y positioning as shown in Figure 4 [40]. Thus, the accuracy value of a GPS data point indicates the probability of the user's true location to be inside the circle centering the measured GPS point and with a radius (or diameter) equal to the reported accuracy value. Note, we consider the uncertainty of horizontal positioning while assuming the time component of the GPS measurement to be accurate, as the time-error is negligible compared to the positioning-error [14].

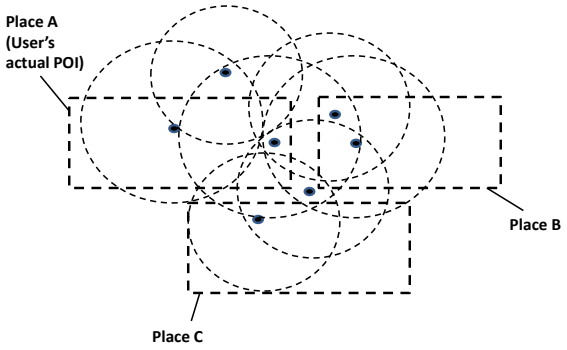

Figure 3: Confidence circles of measured GPS positions

For each measured GPS point, we generate m random (x,y) points inside the corresponding confidence circle as shown in Figure 2 following the bivariate normal distribution [58]:

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp -[\frac{1}{2\sigma^2}\{(x-\mu_x)^2 + (y-\mu_y)^2\}] \quad (1)$$

where (μ_x, μ_y) is the measured (latitude, longitude) and σ is the standard deviation in both directions assuming no correlation between x and y. The value of σ of the above distribution can be obtained by the following equation [58,

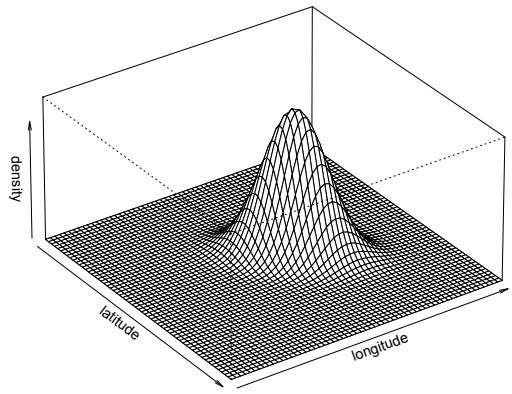

Figure 4: Horizontal position accuracy: GPS fixes are assumed to follow a Bivariate Normal Distribution

59]:

$$\sigma = \frac{r}{\sqrt{(-2 * \log p)}} \tag{2}$$

where p is the probability of the true location to be inside the confidence circle as reported by the GPS device and r is the confidence radius obtained from the reported accuracy value. We attach the same time-stamp to the generated random points within the confidence circle that was used for the measured point which centres the circle.

We perform three dimensional (latitude, longitude and time) kernel density estimation on all the generated random points to estimate their spatio-temporal closeness. The key insight here is that the POI where the user is truly located, should have higher spatio-temporal density of data points than any other nearby POIs. If the user sits at a window side near the border of a building, we can expect the random points to have more spatio-temporal closeness for the actual building due to improved GPS accuracy. On the other hand, if there is no window, the consecutive GPS measurments would rarely overlap each-other due to poor GPS accuracy. However, the generated random points can be expected to have more temporal closeness for the actual building than the nearby POIs. For example, as illustrated on the left hand side of Figure 5, if we consider only the spatial density of the measured GPS location points, the alternative but not visited POI (building B) cannot be differentiated from the user's true POI (building A). However, if we consider the time component as well for density estimation, as shown on the right hand side of Figure 5, the true POI, building A, has greater spatio-temporal density than the pseudo POI. Moreover, if the user stays in multiple nearby buildings in urban canyons, for example in a place like a university, the GPS data points are likely to be more spatio-temporarily sparse for the user's transition paths than for the actual stays. Thus, spatio-temporal density should also help to estimate the spatio-temporarily close events in the user's trajectory, corresponding to her stays at multiple nearby buildings in urban canyons.

Let r_1, r_2,...,r_n be all the generated random points for all the measured GPS points such that $\forall i \in [1,n]$, $r_i = (x_i, y_i, t_i)$. The kernel density estimation of the random points in x, y and t dimensions is given by [57]:

$$\hat{f}_H(r) = \frac{1}{n} \sum_{i=1}^{n} K_H(r - r_i) \tag{3}$$

where

- $r = (x,y,t)^T$,$r_i = (x_i,y_i,t_i)^T$ and $i = 1,2,3,\ldots,n$,
- H is the 3×3 bandwidth matrix,
- K is the kernel function,
- and $K_H(r) = |H|^{-1/2} K(H^{-1/2} r)$.

We use a Gaussian Kernel as K. For d-dimensional data it is defined as

$$K(r) = \frac{1}{\sqrt{(2\pi)^d}} \exp(-\frac{1}{2} r^T r) \tag{4}$$

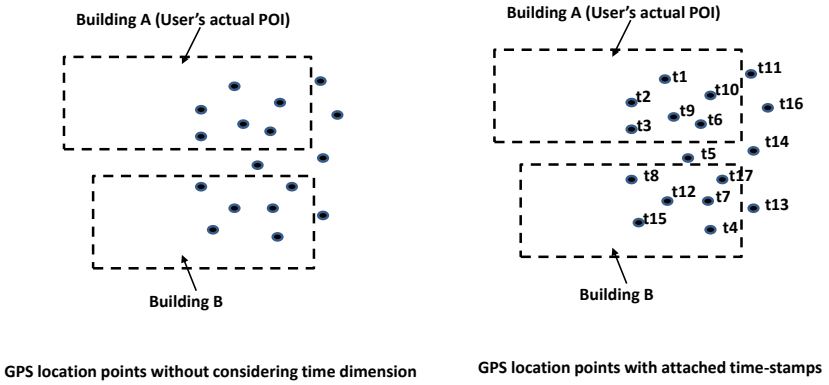

Figure 5: The motivation for using the time dimension for density estimation: The left figure shows the measured GPS positions for a user's stay at a building A. The alternative (but not visited) POI, building B, has almost similar spatial closeness of GPS data points as the actually visited POI. The right figure shows the same location points with attached time stamps. The true POI, building A, has greater temporal closeness of the GPS data points than building B.

In our case we have $d = 3$. The choice of the bandwidth matrix (H) is crucial as it determines the behaviour of the density estimation. A popular strategy for the bandwidth matrix is to select its parameters in order to minimize the Mean Integrated Square Error (MISE) between the estimated density and the true density of the underlying distribution. We have adopted a modified version of the AMISE (Asymptotic Mean Integrated Square Error) technique, developed by Duong et al. in 2003 [60]. This is known as *SAMSE (Sum of Asymptotic Mean Square Error) Pilot Bandwidth Plug-in Estimator* and is available in the R package [61]. The methodology for choosing this technique as the optimal bandwidth estimator is explained shortly in Section 3.3.1.

Instead of estimating the density on the grid points of a spatial grid covering all the data points, kernel estimation assesses density directly on the measured GPS points. This allows us to perform the estimation without depending on the size of individual grid cells. We rank the POIs in descending order of their density weights, with the highest density POI assigned as rank one.

3.3.1. Selecting the optimal bandwidth estimator

The choice of optimal bandwidth matrix is crucial for the performance of kernel density estimation. As illustrated in [62], several methods have been developed for selecting the optimal bandwidth matrix from data. Among them, Cross-Validation techniques such as Least Square Cross Validation (LSCV) [63, 64] or Smooth Cross validation (SCV) [65], the Plug-in estimators [60] and the Normal Scale bandwidth estimators [62] are the most popular for three dimensional data. Both the SCV and Plug-in estimation techniques require an initial pilot bandwidth for optimal bandwidth estimation. Different pilot bandwidths have also been proposed in the literature [62]. We have adopted the SAMSE pilot bandwidth for both SCV and Plug-in bandwidth estimation techniques as proposed by Duong et al. [60]. We also assessed an alternative approach, known as the "Unconstrained" pilot bandwidth [66, 67]), but its use proved infeasible due to the computational complexity.

To select the optimal bandwidth estimator best suited to our type of data, we performed experiments on five different place datasets to check the ranking performance of *Spatio-temporal Density Rank* using different bandwidth estimation techniques. The place datasets corresponded to user's stays in multiple nearby buildings at a University in five different days. The ranking performance was computed in terms of NDCG@5. Please refer to Section 4.3 for detail of this metric. The estimator that produced the highest aggregated NDCG ranking (in terms of mean and median) was then chosen as the optimal bandwidth estimator.

The multi-dimensional versions of cross-validation bandwidth estimators were not directly applicable for our type data since it contains duplicate values in the time dimension. Hence, we also trialled a strategy which estimated the optimal bandwidth in each dimension separately (unique values of latitude, longitude and time) using cross-validation techniques and used a product kernel for density estimation. Table 2 describes the ranking performance of *Spatio-temporal Density Rank* using different optimal bandwidth estimators. Using the SAMSE Plug-in Estimator [60]

produces the best ranking performance with our data and hence this was the method used for all further experiments.

Table 2: Ranking performance of *Spatio-temporal Density Rank* method in terms of mean and median of NDCG@5 values for five different datasets using different optimal kernel density bandwidth estimation techniques. SAMSE Plug-in Estimator has the best ranking performance with our data.

Bandwidth estimation technique	Mean of NDCG@5	Median of NDCG@5
Smoothed Cross Validation (SCV)	0.673	0.622
Least Square Cross Validation (LSCV)	0.673	0.622
Normal Scale (NS)	0.628	0.556
SAMSE Plug-in Estimator	**0.744**	**0.754**

3.4. Second Phase: Ranking POI by Line Segment Intersection (Line Count Rank)

In the first phase, we ranked the POIs according to spatio-temporal density of uncertain GPS data points. However, with real world GPS data, in the worst case, it is possible that the footprint of a user's true POI does not contain any GPS data point. Instead, all the measured GPS positions could be scattered around a building as shown in Figure 6. In such a case, it is difficult to determine if the user is truly located at the POI (in the building) or is just moving around it. We use a *line segment intersection* based approach to differentiate between these two scenarios. The line segments are generated by joining the GPS data points in order of their time sequence. Our main insight is that if the user is actually located at a POI, the number of GPS line segments intersecting that POI will be much higher than if the user is simply moving around it. To support this argument, we compute the expected value of GPS line segments intersecting the POI for both inside and moving around scenarios by extending the classic Buffon-Laplace needle and noodle problem [41, 68]. The Buffon-Laplace needle problem is to find the probability of a needle of length l to cross either of the horizontal or vertical crack when dropped on a floor of tiled rectangles as explained in [41]. Suppose we now bend the needle in any way making it a noodle shape (planar curve). Buffon's noodle problem states that when a noodle of length l is dropped at random on a infinite grid of equally distanced parallel lines, the probability distribution of intersecting a line depends on the shape of the noddle. However, the expected value of line intersection only depends on its length [68]. In our case, we assume a GPS line segment to be a needle whereas a GPS trajectory (multiple line segments joined together in sequence) is a Buffon's noodle. Our theoretical model and simulation results are described in the following sections.

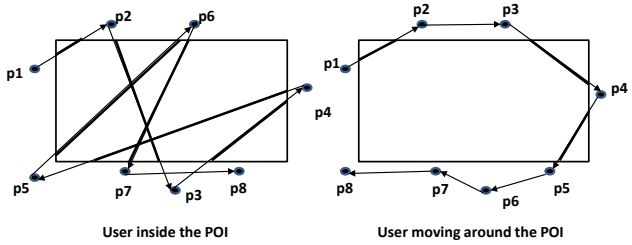

Figure 6: GPS line segments intersecting the POI when the user is inside (the left figure) and moving around (the right figure) the POI. The intersections of line segments with the region are indicated as bold lines. The number of intersections of a user being *inside* or *moving* around the POI are 6 and 3, respectively.

*3.4.1. **A User located inside a POI***

To compute the probability of any GPS line segment intersecting the POI, let us consider the POI to be a rectangle of length a and width b as shown in Figure 7. The measured GPS positions can be outside even when the user is inside the POI due to GPS error. Let us consider an outer rectangle of length $(a+2m)$ and width $(b+2n)$ around the POI such that $[a,b] \gg [m,n]$. For a user located inside the POI, let any two consecutive GPS measurements generate a random line segment $L = \{(p_{x_1}, p_{y_1}) \rightarrow (p_{x_2}, p_{y_2})\}$ of length l such that the base point (p_{x_1}, p_{y_1}) should be $0 \leq p_{x_1} \leq (a+2m)$, $0 \leq p_{y1} \leq (b+2n)$ and the end point (p_{x_2}, p_{y_2}) should be $p_{x_2} \leq (a+2m)$, $p_{y2} \leq (b+2n)$ for $0 \leq \theta \leq 2\pi$. Here θ is the angle made by the line segment with the horizontal axis as shown in the Figure 7. The classic Buffon-Laplace needle problem [41] asks to find the probability $P(l,a,b)$ of a needle of length l to cross at least one horizontal or vertical

crack when dropped on a floor of tiled rectangles of length a and width b, given $l < min(a,b)$. We extend this to work with our scenario assuming $\max(m,n) < l < \min(a,b)$, where l is the length of the GPS line segment in our case.

Theorem 3.1. *Given the length l of a random GPS line segment L, generated when the user is inside the POI, the expectation of the line segment intersecting the POI $e(L)$ is proportional to its length and is given by* $e(L) = \frac{\pi(ab-mn)+2l(m+n)}{\pi(a+m)(b+n)}$.

Proof. Let the distance of the base of the line segment from the next vertical and horizontal boundary of the POI are represented by x and y, respectively as shown in Figure 7. Note that since $0 \leq p_{x_1} < (a+2m)$, the maximum distance the base point can have with any of the vertical boundary of the POI is less than $(a+m)$. Therefore $0 \leq x < (a+m)$. Similarly, we can find $0 \leq y < (b+n)$.

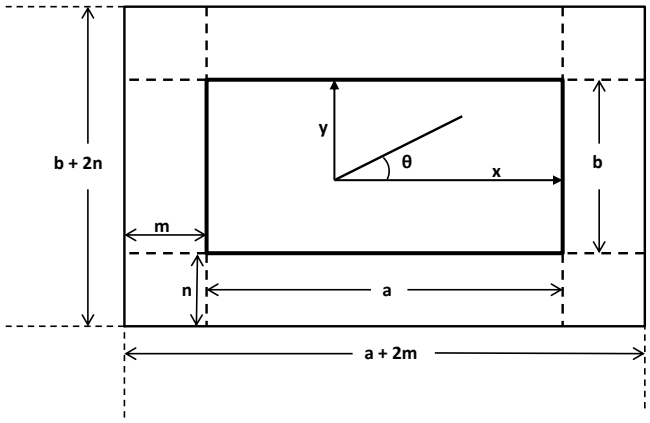

Figure 7: Computing the expectation of a random line segment intersecting the POI considering the user to be inside.

The line segment will intersect the POI if 1) it intersects either a horizontal or vertical POI boundary; or 2) the line segment is completely within the POI. Figure 8 describes the possible positions of the line segment for $\theta < \frac{\pi}{2}$. Similar to Buffon-Laplace needle problem [41], the line segment will intersect a vertical POI boundary if $x < l\cos\theta$. This is represented by the region enclosed by the surface $\theta = arccos(\frac{x}{l})$ and the XY plane for $n \leq y < b+n$. Similarly, the region enclosed by the surface $\theta = arcsin(\frac{y}{l})$ and the XY plane for $m \leq x < a+m$ represents possible positions in which the line segment crosses a horizontal POI boundary. The positions of the line segment that both a horizontal and a vertical intersection is represented by the intersection of these two regions. As shown in Figure 8, the projection of this intersection on the XY plane is a circle, where $\arcsin(\frac{y}{l}) = \arccos(\frac{x}{l})$ implying $x^2+y^2 = l^2$. However, the line segment will intersect both the horizontal and the vertical POI boundary only when $x \geq m$ and $y \geq n$ represented by the bold arc in Figure 8. Thus, the condition for intersecting both the horizontal and the vertical POI boundary is given by the equation

$$\begin{array}{rcl} m & \leq & x < l\cos\theta \\ n & \leq & y < l\sin\theta \end{array} \tag{5}$$

If we assume that x,y and θ are uniformly distributed within their respective ranges then the probability of a vertical intersection (P_{vc}) can be computed as

$$\begin{array}{rcl} P_{vc} & = & \dfrac{\text{volume for which the line segment intersects a verical POI boundary}}{\text{total volume of the domain}} \\ & = & \dfrac{\int_0^{\frac{\pi}{2}} \int_n^{b+n} \int_0^{lcos\theta} \mathrm{d}x\, mathrmdy\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} \end{array} \tag{6}$$

Similarly, the probabilities of a horizontal intersection (P_{hc}) and a vertical and horizontal intersection ($P_{hc\cap vc}$) are given by the following equations

$$P_{hc} = \frac{\int_0^{\frac{\pi}{2}} \int_m^{a+m} \int_0^{lsin\theta} \mathrm{d}y\mathrm{d}x\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} \tag{7}$$

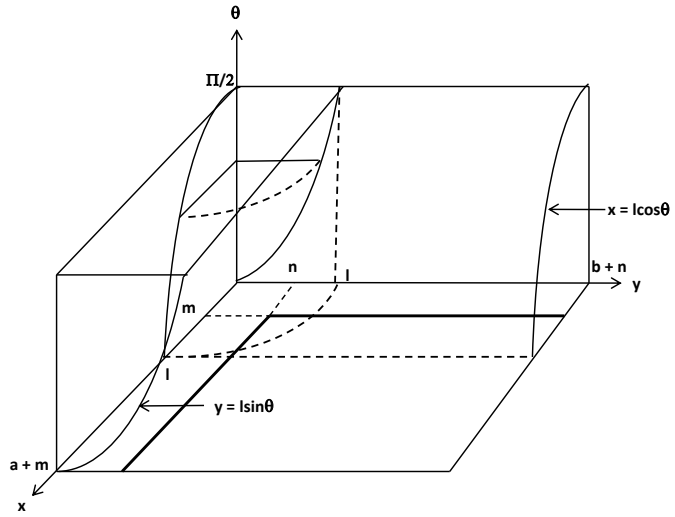

Figure 8: 3-dimensional view of possible positions of a random line segment for $\theta < \frac{\pi}{2}$ considering a user to be inside

$$P_{hc\cap vc} = \frac{\int_0^{\frac{\pi}{2}} \int_m^{lcos\theta} \int_n^{lsin\theta} \mathrm{d}y\, mathrmdx\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} \tag{8}$$

Now, the probability of either a horizontal or vertical intersection can be computed as

$$P_{hc\cup vc} = P_{hc} + P_{vc} - P_{hc\cap vc} \tag{9}$$

Integrating and solving the equations leads to

$$P_{hc\cup vc} = \frac{2l(a+b+m+n) - l^2 - \pi mn}{\pi(a+m)(b+n)} \tag{10}$$

In the case $m = n = 0$ then this is similar to the classic Buffon-Laplace needle problem [41], where the probability of either a horizontal or vertical intersection is given by

$$P_{hc\cup vc} = \frac{2l(a+b) - l^2}{\pi ab} \tag{11}$$

The line segment will be completely inside the POI for $\theta < \frac{\pi}{2}$, if

$$\begin{array}{rcl} m & \leq & x < a+m-l\cos\theta \\ n & \leq & y < b+n-l\sin\theta \end{array} \tag{12}$$

Therefore, the probability of the line segment to be completely inside the POI boundary can be obtained as

$$P_{in} = \frac{\int_0^{\frac{\pi}{2}} \int_m^{a+m-lcos\theta} \int_n^{b+n-lsin\theta} \mathrm{d}y\mathrm{d}x\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} = \frac{\pi ab - 2l(a+b) + l^2}{\pi(a+m)(b+n)} \tag{13}$$

Note that there will be only one line segment intersection irrespective of whether it 1) crosses either a horizontal or vertical boundary or 2) it remains completely inside the POI. Therefore, we can compute the expectation of line intersection from equations 10 and 13 as follows.

$$e(L) = P_{hc\cup vc} + P_{in} = \frac{\pi(ab-mn) + 2l(m+n)}{\pi(a+m)(b+n)} \tag{14}$$

In the above equation if a, b, n and m remain constant, then we can say that the expected value of a line segment intersecting the POI will be proportional to its length. The higher the length of a line segment the higher will be the expected value of intersections with the POI.

□

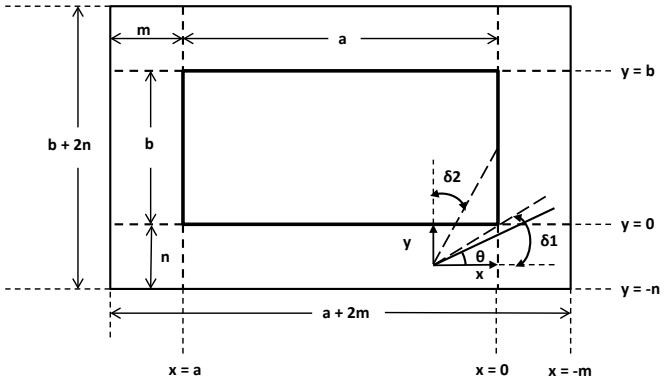

Figure 9: Computing the expectation of a random line segment intersecting the POI if the user moves around the POI.

3.4.2. ***Moving around the POI***

Similarly, we can compute the expected value of a line segment intersecting the POI when the user is moving around it. Let us again consider a rectangular POI of length a and width b and an outer rectangle of length $(a+2m)$ and width $(b+2n)$ around the POI such that $[a,b] \gg [m,n]$. This is shown in Figure 9. We assume for a user moving around the POI, any two consecutive GPS measurements will generate a random line segment in such a way that no end point of the segment will penetrate inside the POI. Let (p_{x_1}, p_{y_1}) and (p_{x_2}, p_{y_2}) be the base and end point of the line segment, respectively. We assume that though p_{x_1} can vary in the range of $0 \leq p_{x_1} \leq (a+2m)$, however when $m \leq p_{x_1} \leq (a+m)$ then $0 \leq p_{y_1} \leq n$ or $(b+n) \leq p_{y_1} \leq (b+2n)$. Similarly, though p_{y_1} can vary in the range of $0 \leq p_{y_1} \leq (b+2n)$, however when $n \leq p_{y_1} \leq (b+n)$ then $0 \leq p_{x_1} \leq m$ or $(a+m) \leq p_{x_1} \leq (a+2m)$. A similar logic applies to the variation of p_{x_2} and p_{y_2} as well where $p_{x_2} \leq (a+2m)$ and $p_{y_2} \leq (b+2n)$, for $0 \leq \theta \leq 2\pi$. Here θ is the angle made by the line segment with the horizontal axis. Clearly, in such scenario a line segment can intersect the POI (the inner rectangle) only at the four corners. Similar to Theorem 3.1, let us assume l be the length of the line segment such that $\max(m,n) < l < \min(a,b)$.

Theorem 3.2. *Given the length l of a random GPS line segment L, generated when the user is moving around the POI, the expectation of the line segment intersecting the POI $e(L)$ is given by* $e(L) = \dfrac{l^2 + 2l(m+n) + \pi mn}{\pi(a+m)(b+n)}$.

Proof. The distance of the base of the line segment from the next vertical and horizontal boundary of the POI are represented by x and y respectively. Note that since the user is moving around the POI, for $0 \leq \theta < \frac{\pi}{2}$, x and y can vary in the following way: $-m \leq x < a$ and $-n \leq y < b$. Therefore the total volume of the domain in which x,y and θ can vary is $\frac{\pi}{2}(a+m)(b+n)$.

The line segment will intersect the POI only when there is both vertical ($x < l\cos\theta$) and horizontal ($y < l\sin\theta$) intersection. However since the user moves around, no end point of the line segment can lie inside the POI. Therefore, the range of θ, for which there are a horizontal and a vertical intersection with the POI boundary, is $\delta_1 \leq \theta < (\frac{\pi}{2} - \delta_2)$, where δ_1 and δ_2 are greater than 0. This is described in Figure 9. Hence the condition for both a horizontal and a vertical intersection is:

$$\begin{array}{rcl} -m & \leq & x < l\cos\theta \\ -n & \leq & y < l\sin\theta \\ \delta_1 & \leq & \theta < (\frac{\pi}{2} - \delta_2) \end{array} \tag{15}$$

Therefore the probability of a line segment intersecting the POI is

$$P_{hv} = \frac{\int_{\delta_1}^{\frac{\pi}{2}-\delta_2} \int_{-m}^{l\cos\theta} \int_{-n}^{l\sin\theta} \mathrm{d}y\mathrm{d}x\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} \tag{16}$$

Let us assume that $l \gg m,n$ so that $[\delta_1, \delta_2] \approx 0$. This assumption agrees with GPS-enabled mobile devices, since they have large measurement errors specially in urban canyons. Substituting δ_1 and δ_2 with 0 in the above equation we

find

$$P_{hv} = \frac{\int_0^{\frac{\pi}{2}} \int_{-m}^{l\cos\theta} \int_{-n}^{l\sin\theta} mathrmdy\mathrm{d}x\mathrm{d}\theta}{\frac{\pi}{2}(a+m)(b+n)} = \frac{2l(m+n)+l^2+\pi mn}{\pi(a+m)(b+n)} \tag{17}$$

Since there is only one line intersection, the expectation of a random line segment intersecting the POI, when the user moves around it, is

$$e(l) = P_{hv} = \frac{l^2+2l(m+n)+\pi mn}{\pi(a+m)(b+n)} \tag{18}$$

Assuming a, b, m, n to be constants in the above equation, we can say that when the user is moving around the building the expectation of a random line segment intersecting the POI is proportional to the square of its length. □

*3.4.3. **Expectation of the number of line segments of a trajectory intersecting the POI***

In the previous sections, we computed expectations of a random line segment intersecting the POI assuming user's trajectory consisting of only two GPS points. However, normally a trajectory contains multiple GPS points with attached time-stamps. If we join the GPS points according to their time sequence, we obtain multiple line segments joined one after another in a sequence similar to Buffon's noodle [68]. To compute the expectation of line segments of a trajectory intersecting the POI, let us consider a trajectory L consisting of n line segments $L_1, L_2, L_3, \ldots, L_n$. Let l be the length of the entire trajectory such that $l = \sum_{i=1}^{n} l_i$, where l_i is the length of the ith line segment of the trajectory. Similar to Buffon's noodle problem [68], the probability distribution of the number of line intersections the POI depends on the shape of the trajectory but the expected value depends only on its length.

Proposition 1. *Given a trajectory L consisting of n line segments $L_1, L_2, L_3, \ldots, L_n$, the expected value of the number of line segments intersecting the POI is given by $e(L) = \sum_{i=1}^{n} e(L_i)$, where $e(L_i)$ is the expected value of line intersections of each individual segment L_i.*

Proof. Similar to the Buffon's noodle problem, the above can be proved by induction on n once the case of $n = 2$ is established [68]. Let us consider the trajectory (L) to be consisted of two line segments L_1 and L_2. The events for which there are line intersections with the POI are as follows:

- Event A: Only L_1 intersects the POI
- Event B: Only L_2 intersects the POI
- Event C: Both L_1 and L_2 intersect the POI

Let P_A, P_B and P_C represent the probabilities of events A, B and C respectively. Since there is only one line intersection for the events A and B and two line intersections for event C, therefore the expectation of line intersections of L can be written as

$$e(L) = 1 * P_A + 1 * P_B + 2 * P_C = (P_A + P_C) + (P_B + P_C) \tag{19}$$

Now, since L_1 will intersect the POI only for events A and C, therefore the expectation of line segment L_1 intersecting the POI is given by $e(L_1) = P_A + P_C$. Similarly, the expectation of line segment L_2 intersecting the POI is $e(L_2) = P_B + P_C$. Replacing the values of $e(L_1)$ and $e(L_2)$ in the above, we obtain

$$e(L) = e(L_1) + e(L_2) \tag{20}$$

Thus by induction on n, we can say that the expectation of line segments intersecting the POI for the entire trajectory is the sum of the expectations of line intersecting of individual segments. □

Corollary 1. *Let us consider two trajectories, each consisting of n number of line segments generated randomly when the user is inside the POI and moving around it, respectively. Let each corresponding segment in the two trajectories are of same length. Then the expected value of the number of line segments intersecting the POI is higher when the user is considered to be inside than the moving around scenario.*

Proof. Similar to Theorem 1, the above corollary can be proved by induction once the case of $n = 2$ is solved. Let L_{in} be the trajectory consisting of two line segments $L_{A_{in}}$ and $L_{B_{in}}$ generated randomly of a user located inside the POI. Let us consider another trajectory L_{out} consisting of two line segments $L_{A_{out}}$ and $L_{B_{out}}$ generated randomly when the user moves around the POI. Let l_a be the length of the line segments $L_{A_{in}}$ and $L_{A_{out}}$ and l_b be the length of the line segments $L_{B_{in}}$ and $L_{B_{out}}$.

From Theorem 3.1 and 1 follows that the expected value of the number of line segments intersecting the POI of a user located inside it, is

$$e(L_{in}) = e(L_{A_{in}}) + e(L_{B_{in}}) = \frac{\pi(ab-mn)+2l_a(m+n)}{\pi(a+m)(b+n)} + \frac{\pi(ab-mn)+2l_b(m+n)}{\pi(a+m)(b+n)} \tag{21}$$

Similarly, from Theorem 3.2 and Theorem 1 follows that the expected value of the number of line segments intersecting the POI of a user moving around it, is

$$e(L_{out}) = e(L_{A_{out}}) + e(L_{B_{out}}) = \frac{l_a^2+2l_a(m+n)+\pi mn}{\pi(a+m)(b+n)} + \frac{l_b^2+2l_b(m+n)+\pi mn}{\pi(a+m)(b+n)} \tag{22}$$

Equation 21 and 22 show that $e(L_{in}) > e(L_{out})$ only when $2\pi(ab-mn) > l_a^2 + l_b^2 + 2\pi mn$. Now, since according to our assumptions $[m,n] < l_a < [a,b]$, $[m,n] < l_b < [a,b]$ and $[a,b] \gg [m,n]$, therefore we can say $e(L_{in}) > e(L_{out})$. Thus by induction on n we can say that the expected value of the number of line segments intersecting the POI is higher when the user is considered to be inside the POI than the moving around scenario. □

3.4.4. ***Simulation Results***

To evaluate the practical utility of Theorems 3.1 and 3.2 from Sections 3.4.1 and 3.4.2, respectively, we performed experiments to simulate both *inside* and *moving around* scenarios. We assume $a = 30$ meters, $b = 20$ meters and $m = n = 5$ meters (the values for a and b reflect the size of a typical office building). The length (l) of a random line segment is varied such that $5 < l < 20$ meters. For a particular value of l, we run both the experiments separately for 5000 iterations. The simulated mean is computed as *(the number of iterations for which a random line segment intersects the POI)/(total number of iterations)*. Figure 10 shows the expected and simulated average line intersections with varying length of the line segment. Clearly, the expectations (or simulated average) of line intersections increase with increasing of length of line segment in both *moving around* and *inside* scenarios. However, for a particular length, the expected value is much higher when the user is considered to be inside than when he or she is assumed to be moving around the POI.

To evaluate Proposition 1 in Section 3.4.3, we also performed a simulation experiment to show that the expected values are indeed additive in a practical scenario when considering an entire trajectory. We assume that the trajectory consists of two line segments (L_1 and L_2). We assume $a = 30$, $b = 20$ meters and $m = n = 5$ meters. Considering the user to be inside the POI, the length of the line segments l_1 and l_2 are varied such that $5 < l_1, l_2 < 20$ meters. For a particular length ($l = l_1 + l_2$) of the trajectory, it is randomly thrown on the POI for 5000 times. The expected and simulated average line segment intersecting the POI is shown in Figure 11. Clearly, the simulated line intersections are very close to the expected values for each l, thus showing that Theorem 1 is applicable in more practical scenarios.

Usage of theoretical model: Hence, in the second phase of our algorithm (*POI-ID*), we use the number of line segments intersecting a POI as a measure to check if the user is inside it. Each POI is given a weight according to the number of GPS line segments intersecting that POI. We then rank the POIs in the descending order of their *line-count* weight. The POI with the highest *line-count* is given rank one.

3.4.5. Mean Reciprocal Ranking

We use the Mean Reciprocal Rank (MRR) to combine the rankings of each POI from the previous two phases. The MRR is a well known ranking mechanism that is widely used in Information Retrieval. The MRR of a process that produces possible responses to a set of sample queries Q is given by [69]

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{23}$$

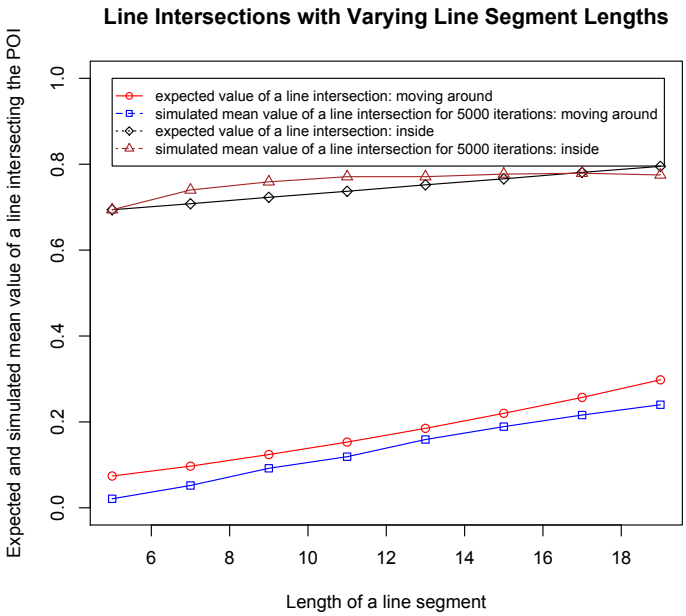

Figure 10: Variation of expectation and simulated average line segment intersecting the POI with increasing segment length in moving around and inside scenarios.

In our case, $|Q| = 2$, since we have two different phases (or queries) to rank the POIs. Therefore, each POI is given a mean reciprocal rank as follows -

$$MRR_{poi} = \frac{1}{2}\left(\frac{1}{rank_{density}} + \frac{1}{rank_{linecount}}\right) \tag{24}$$

where $rank_{density}$ and $rank_{linecount}$ are the density rank and the line-count rank of the POI, respectively.

4. Experimental Evaluation

4.1. Data Collection

We have evaluated our algorithm on real world user GPS data. The publicly available GPS datasets [70–72] generally do not contain ground-truth mainly due to privacy considerations. Since we need ground truth to validate our results, we collected our own data using an application on an android mobile phone (Samsung Galaxy SII) and a tablet (Nexus 7). The android phone had an in-built assisted GPS receiver and the tablet had an ordinary in-built GPS unit. The data was collected from two users studying at a university in Melbourne, Australia. The users recorded their GPS locations while staying at different significant indoor places, such as home, university, childcare, kindergarten and shopping malls. They also tagged a place by entering the place name, street address and time of entry through the mobile application. These annotations served as ground truth to validate our result. There are many factors affecting the GPS accuracy at indoor places, for example building elements and materials, number of floors and walls, density of surrounded buildings and their relative sizes, etc. [15]. Hence we sampled our data from a range of variety of different places with varying GPS accuracy to make our algorithm work in real world scenario. The university is located in a central area of Melbourne and is representative of a location with densely populated POIs or buildings within a small region. The users stayed in different buildings (POIs) during their university stays: office buildings, cafeteria, lecture theatres and other nearby buildings. The buildings were mainly built out of concrete with regular roofs, walls, windows and multiple floors. The users stayed on different floors of the buildings varying from the 1st to 10th. On the other hand, the users' homes were located in a residential area containing sparsely populated POIs. The childcare and kindergarten were also located in outer suburbs, but they were places where less time was spent (less than five minutes, to drop off or pick up kids). The shopping mall was located in a wide region in the outer suburbs.

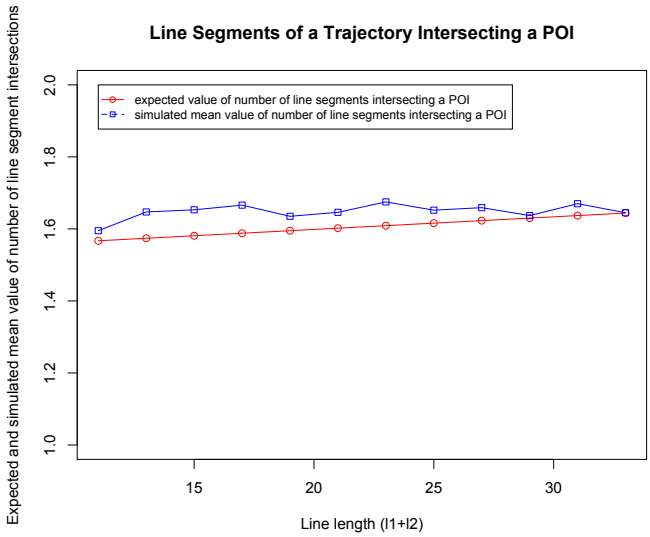

Figure 11: Variation of the expectation and the simulated average trajectory intersections with the POI with increasing lengths. A trajectory L consists of two randomly generated line segments (L_1 and L_2) for a user inside the POI.

It had multiple spread buildings with glass-roof over multiple floors. The POIs in the shopping mall were categorized at building levels.

We ran our experiment on a total of 100 datasets collected over a 3 month period. Each dataset represented *place data* as defined in Section 3.1. The android application was set to collect and record data at every 10 seconds interval. Each dataset contains on average 429 GPS points with the smallest dataset having 10 and the largest dataset having 1816 points. The size of the total dataset is comparable to earlier literature [20, 47, 48]. Table 3 describes the number of datasets of different place types on which we run our experiment. Approximately 60% of the places in our dataset were collected at the university, since it represents the location in urban canyons with poorest GPS accuracy, 40 to 70 meters. 23% of our data were childcare/kindergarten locations where the user stayed only briefly. The remaining 18% represents locations where the GPS accuracy was below 20 meters. We have created our own POI dataset (as defined in Section 3.1) using Google Maps [73] and Openstreet Map [74].

Table 3: Number of location files of different location types

Location Type	Number
University	59
Childcare	19
Kindergarten	4
Home	16
Shopping Mall	2

4.2. Baseline Techniques

The existing literature is not directly related to our work. Previous work, which considered only GPS data, focused on identifying places at a much coarser level, for example, at the level of a university, rather than on individual buildings. Please refer to the Table 1 in Section 2 for details. In order to compare the performance of our algorithm with the existing approaches, we first identify the GPS clusters and the cluster centers (*stay points*) using the existing algorithms. Next we map each *stay point* to its nearest building (POI). POIs are then ranked according to the duration of stay, with the largest duration POI as rank one and so on. We compared our work with the state-of-the-art place identification techniques (baselines 1-3) as proposed by Zheng et al. [12, 33, 34, 45], Palma et al. [32] and Bhattacharya et al. [36] which like ours use only GPS data. We have also constructed our own competing baseline techniques (baselines 4-7) to serve as a reference for comparing the performance of our algorithm. All the baseline

techniques are implemented on the cleaned datasets after preprocessing them as explained in Section 3.2.

Baseline 1: Zheng et al.. We compare our method with the place extraction technique as proposed by Zheng et al. [33] and afterwards followed in a series of other works [12, 34, 45]. A stay point (S) has been regarded as a virtual location and extracted as a cluster of consecutive GPS points, which are bounded by a maximum distance threshold (D_{th}) and a minimum time duration (T_{th}). The location of the *stay point* (S_x, S_y) has been computed as the mean of the extracted GPS points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ by the following equation:

$$S_x = \sum_{i=1}^{n} x_1/n, \qquad S_y = \sum_{i=1}^{n} y_1/n \tag{25}$$

Setting D_{th} as 200 meters and T_{th} as 20 minutes as in [33], returned a single cluster with our datasets. Hence to work with our data, specifically with the university datasets where multiple buildings are closely located within a small region, we set D_{th} as 100 meters and time threshold T_{th} as 5 minutes.

Baseline 2: Palma et al.. We compare our method with the place extraction technique as proposed by Palma et al. [32]. A variation of DBSCAN algorithm has been used to identify the low speed segments from a user's trajectory as significant places. The *Eps* parameter has been estimated based on a quantile function as proposed in [32]. The *MinTime* parameter has been set to 120 secs, reflecting the minimum time required to consider the GPS points to be in a same cluster.

Baseline 3: Bhattacharya et al.. We compare the performance of POI-ID with our prior work [36] where clusters of consecutive GPS points have been extracted as significant places based on user speed, acceleration and bearing change threshold. The low speed segments of a trajectory, for which the standard deviations of bearing changes are beyond a threshold value, have been extracted as significant places. The *stay points* for the extracted segments have been computed by the equation 25 as in [33]. We set the threshold values similar to [36].

Baseline 4: Random-rank. We compare our method against a random ranking algorithm. Given a POI dataset of m nearby POIs, a random ranking algorithm will assign a rank to each POI by choosing a unique random number from 1 to m.

Baseline 5: Distance-to-staypoint. In prior literature the so-called *stay point* (S_x, S_y) has been computed as the mean of latitude and longitude of all the GPS points, $l_1, l_2, l_3, \ldots, l_n \in L$ corresponding to a user's stay at a place as shown in equation 25 [33, 34, 36, 45**?**]. However, it is well known that a *stay point* cannot be directly mapped to a POI due to high GPS noise at indoor places specially in urban canyons [75]. A common approach is to rank the POIs based on the distance from the stay-point. We chose this *Distance-to-staypoint* technique as baseline to compare it with our algorithm for highly noisy GPS data.

Baseline 6: Density-rank. In this technique we consider only the spatial closeness of measured GPS points ignoring the temporal density. We rank the POIs according to the kernel density estimation of the GPS points in two dimensions (latitude, longitude).

Baseline 7: Point-Count-Rank. In this method we assume the measured GPS positions to be perfectly accurate. We rank the POIs according to the number of GPS points contained in that POI.

4.3. Evaluation Metrics

We present the metrics we used to evaluate the performance of *POI-ID*.

Precision@1. For a given location query *Precision*@1 is given by

$$Precision@1 = rel_1 \tag{26}$$

where $rel_1 \in \{0, 1\}$ indicates if the location retrieved at position 1 in the result set is the user's true location.

Precision@k. For a given location query, *Recall@k* for the top k locations in the result set is computed as

$$Recall@k = \frac{1}{n}\sum_{i=1}^{k} rel_i \tag{27}$$

where n is the number of user's true locations. $rel_i \in \{0,1\}$ indicates if the location at position i in the result is the user's true location.

Normalized Discounted Cumulative Gain (NDCG). Recall is a set based metric. It cannot differentiate between an ordered and unordered result set. Our aim is to retrieve a user's true POIs in an ordered manner. For example if a user stays primarily at a university building A and takes lunch in another building say, B; then in an ideal result set A should be at position 1 and B at position 2. To evaluate the ranking performance of our algorithm, we have used NDCG [76], which is a widely used measure in information retrieval systems to assess the ranking performance of search engines. In our case, for a given location query the NDCG of the ranking for the top k retrieved locations is:

$$nDCG@k = \frac{1}{Z}\sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)} \tag{28}$$

where Z is the normalization factor such that an ideal ranking upto position k produces the NDCG value of 1.

4.4. Results and Discussion

Table 4 shows the average *precision*@1 for the rankings generated by *POI-ID* and the baseline approaches for different place types. We can see that the precision@1 *Palma et al.* and *Random-rank* are very poor for all location types. *Palma et al.* has similar performance as *Random-rank* for shopping mall and home datasets, but has worse performance for others. It cannot identify any cluster for any of the kindergarten datasets due to very few data points captured in those places. *Palma et al.* implements a speed based DBSCAN algorithm to identify clusters of GPS points corresponding to low speed segments from a user's trajectory. Since 95% of our data points have been collected at indoor places, the speed based clustering technique did not work well with our data. The baselines *Distance-to-staypoint*, *Zheng et al.* and *Bhattacharya et al.* have very poor accuracy for university locations due to the surrounding tall buildings. Moreover, in a university a user can stay in multiple sub-places, for example attending class in a lecture theatre, having lunch in a canteen or reading in a library. Thus, assuming the measured GPS points represent a Gaussian model around a particular location (mean coordinate) as in *Distance-to-staypoint* is not effective. All baseline techniques except the *Palma et al.* and *Random-rank* are equally good as *POI-ID* in terms of their *Precision@1* value for home and shopping mall datasets due to good accuracy of the GPS measurements at those places. In fact, our previous work *Bhattacharya et al.* [36] has better *Precision@1* value for home locations though has poor performance for university datasets. *Bhattacharya et al.* extracts a cluster of consecutive GPS points as a place based on the bearing change distribution. The rationale is that when a user is driving or walking on a road his bearing is restricted by the direction of the road. On the contrary, if the user is staying at a place the bearing can be any value between $0-360$ degrees, either due to poor GPS accuracy or the user's movement in that place. Since in a place like university, a user can move from one building to another without following a directed road and, thus, the bearing change based approach does not help much in identifying user's POI(s) at building levels.

Our proposed algorithm (*POI-ID*) outperforms the existing approaches for all location types. It also outperforms our proposed competitive baseline techniques for locations with unreliable GPS measurements such as the childcare and university datasets. The childcare locations are places where one user spent only few minutes to drop off or pick up the kids. Therefore during this short time it may not have been possible to obtain any data point inside the actual POI. Since the baseline techniques do not consider the inaccuracy of GPS measurements, for some childcare datasets they could not estimate the true POI. Similarly, university datasets represent locations surrounded by tall buildings in the city and in such circumstances *POI-ID* performs better than all the baseline techniques. Interestingly, all the baseline techniques except the *Palma et al.* and *Random-rank* perform equally well to *POI-ID* for the kindergarten datasets, despite the fact that the nature of the place is quite similar to that of the childcare center. However, since the kindergarten location was situated beside a park with no surrounding nearby POIs, obtaining the true POI was

easier for the kindergarten datasets than the childcare datasets. The overall *precision@1* values for all 100 datasets are shown in Table 5.

Table 4: Average precision@1 of *POI-ID* and the baseline techniques for estimating user's true POI(s) during his or her stay at different type of significant places. The dashed lines separate the existing methods, our constructed competing baseline techniques and the proposed algorithm. *POI-ID* outperforms all the baseline techniques specifically for place types with unreliable GPS measurements like a university or a childcare center.

	Methodology	University	Childcare	Kindergarten	Shopping mall	Home
Existing baselines	Palma et al.	0.017	0.210	0	0.500	0.250
	Zheng et al.	0.085	0.789	1	1	0.938
	Bhattacharya et al.	0.068	0.842	1	1	1
Proposed competing baselines	Random-rank	0.051	0.420	0.750	0.500	0.250
	Distance-to-staypoint	0.034	0.789	1	1	0.938
	Density-rank	0.271	0.842	1	1	0.938
	Point-count-rank	0.288	0.842	1	1	0.938
Proposed algorithm	POI-ID	**0.305**	**0.895**	1	1	0.938

Table 5: Overall precision@1 of *POI-ID* and the baseline techniques for all 100 location datasets. The dashed lines separate the existing methods, our constructed competing baseline techniques and the proposed algorithm

	Methodology	Precision@1
Existing baselines	Palma et al.	0.100
	Zheng et al.	0.410
	Bhattacharya et al.	0.440
Proposed competing baselines	Random-rank	0.189
	Distance-to-staypoint	0.380
	Density-rank	0.530
	Point-count-rank	0.540
Proposed algorithm	POI-ID	**0.560**

Table 6 describes the average recall across all location datasets at different rank cut-offs produced by *POI-ID* and the baseline techniques. *POI-ID* outperforms all the baseline approaches, although the recall@2 and recall@3 of *Point-Count-Rank* or *Density-Rank* are close to *POI-ID*. Note that, the recall values of the existing baseline techniques (*Palma et al.*, *Zheng et al.* and *Bhattacharya et al.*) do not improve much with higher rank cut-offs ($k \geq 3$) since the algorithms have not been designed to estimate and rank POI(s) at building levels. The average recall for selecting the top 5 and top 3 locations of *POI-ID* are 96.5% and 84.17%, respectively.

Table 6: Average recall performance of *POI-ID* and the baseline techniques at different rank cut-offs (k values) for all 100 datasets. The dashed lines separate the existing methods, our constructed competing baseline techniques and the proposed algorithm. *POI-ID* outperforms all the baseline techniques in estimating a user's true POI(s) within the top k retrieved locations.

	Methodology	k=1	k=2	k=3	k=4	k=5
Existing baselines	Palma et al.	0.100	0.140	0.160	0.180	0.185
	Zheng et al.	0.373	0.413	0.448	0.458	0.468
	Bhattacharya et al.	0.388	0.403	0.423	0.423	0.423
Proposed competing baselines	Random-rank	0.163	0.268	0.421	0.557	0.647
	Distance-to-staypoint	0.348	0.452	0.532	0.597	0.952
	Density-rank	0.483	0.677	0.832	0. 910	0.940
	Point-count-rank	0.493	0.682	0.837	0.905	0.950
Proposed algorithm	POI-ID	**0.513**	**0.697**	**0.842**	**0.915**	**0.965**

Our algorithm obtains significant advantages over all the baseline techniques when considering the ranking or ordering performance for the retrieved locations. The ranking performance is computed in terms of NDCG@3 and NDCG@5 to determine how *POI-ID* can rank the user's true POIs within the top 3 and top 5 retrieved locations.

Figure 12 describes the average and median NDCG values of all location datasets as obtained by *POI-ID* and the baseline techniques. The bold line in each box describes the median value of the corresponding technique. The points and texts demonstrate the average values. We can see that *POI-ID* outperforms all the techniques in terms of both NDCG@3 and NDCG@5.

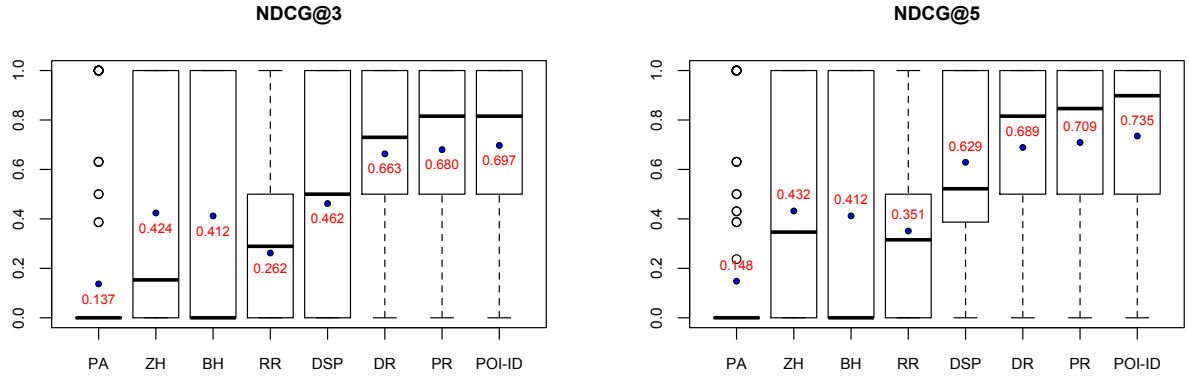

Figure 12: Overall ranking performance of *POI-ID* and the baseline techniques for all 100 datasets in terms of NDCG@3 and NDCG@5. The abbreviations of the baseline techniques are as follows: PA= Palma et al., ZH=Zheng et al., BH=Bhattacharya et al., RR=Random-Rank, DSP=Distance-to-staypoint, DR=Density-Rank,PR=Point-Count-Rank. The bold line in each box describes the median value of the corresponding technique. The blue points and red texts demonstrate the average values. *POI-ID* performs better than the baseline techniques to rank the user's true POI(s) within the top 3 and 5 retrieved locations.

The result of a significance test (*Wilcoxon Signed Rank Test*) in terms of the NDCG@5 while comparing *POI-ID* against the baseline approaches is summarized in Table 7. The confidence interval was set at 95% . The different null hypothesis are also described in the table. In all comparisons we obtain p-values either to be $\ll 0.05$ or < 0.05, thus obtaining significant performance improvement of our method against all the baseline techniques.

Table 7: Statistical significance test (Wilcoxon Signed Rank Test): Comparing ranking performance *POI-ID* and the baseline techniques in terms of NDCG@5 for each of the 100 datasets. The dashed line separates the existing methods from our constructed baseline techniques. The confidence interval was set at 95%. Thus, a p value < 0.05 indicates the null hypothesis is statistically unlikely and hence can be rejected.

Null Hypothesis	p-value
POI-ID and *Palma et al.* have similar NDCG@5	3.85e-15
POI-ID has lesser NDCG@5 than *Palma et al.*	1.92e-15
POI-ID and *Zheng et al.* have similar NDCG@5	2.70e-10
POI-ID has lesser NDCG@5 than *Zheng et al.*	1.35e-10
POI-ID and *Bhattacharya et al.* have similar NDCG@5	6.48e-10
POI-ID has lesser NDCG@5 than *Bhattacharya et al.*	3.24e-10
POI-ID has lesser NDCG@5 than *Density-Rank*	9.00e-04
POI-ID and *Random-Rank* have similar NDCG@5	1.84e-13
POI-ID has lesser NDCG@5 than *Random-Rank*	9.20e-14
POI-ID and *Distance-to-staypoint* have similar NDCG@5	2.00e-04
POI-ID has lesser NDCG@5 than *Distance-Staypoint-Rank*	1.00e-04
POI-ID and *Point-Count-Rank* have similar NDCG@5	2.30e-02
POI-ID has lesser NDCG@5 than *Point-Count-Rank*	1.10e-02
POI-ID and *Density-Rank* have similar NDCG@5	2.00e-03
POI-ID has lesser NDCG@5 than *Density-Rank*	9.00e-04

4.4.1. Analysing the Components of POI-ID

We performed further experiments to analyse the ranking performance of individual components of *POI-ID*. Table 8 summarizes the average NDCG@3 and NDCG@5 for all location datasets of each individual component of *POI-ID*. Comparing against the performance of the baseline techniques as in Figure 12, we can find the *Line-count-rank* component outperforms all the baseline approaches in terms of NDCG@3 and NDCG@5 whereas the *Spatio-*

temporal-density-rank component outperforms all the baseline techniques except the *Point-count-rank* technique. We found *Spatio-temporal-density-rank* has specifically poor performance compared to *Point-count-rank* technique for our university datasets. This is because GPS noise for some university datasets was also spatially and temporarily close as the actual points. Thus the *Spatio-temporal-density-rank* technique could not differentiate the GPS data points corresponding to the user's true POI from the nearby *pseudo* POIs. However, on the university datasets for which the user visited multiple sub-places, for example nearby buildings to attend seminars or to take lunch, *Spatio-temporal-density-rank* outperforms all the baseline techniques including the *Point-count-rank* in terms of both NDCG@3 and NDCG@5. As shown in Table 8, the best ranking performance is obtained by combining *Spatio-temporal-density-rank* with *Line-count-rank* as in *POI-ID*. We also obtain significant performance improvement in terms of NDCG@5 for *POI-ID* while comparing against its individual components (p value < 0.05).

Table 8: Overall ranking performance of the components of *POI-ID* for all 100 datasets in terms of NDCG@3 and NDCG@5.

Methodology	NDCG@3	NDCG@5
Spatio-temporal density rank (phase 1 of POI-ID)	0.668	0.703
Line count rank (phase 2 of POI-ID)	0.687	0.729
POI-ID	0.697	0.735

Table 9 summarizes the ranking performance of different techniques for university datasets containing a single place (user stayed at a single building) and containing sub-places (user stayed at different buildings at the university). Although, *Spatio-temporal-density-rank* component has poor performance compared to *Point-count-rank* for *single place* datasets, it outperforms all the baseline techniques for the subplace datasets. In such datasets, considering the temporal dimension has helped to estimate the spatio-temporarily close events corresponding to the user's stays at different nearby buildings (sub-places). In addition, due to poor accuracy in urban canyons, GPS measurements are often in the nearby buildings during a user's movement from one building to another, instead of the actual transition path. As a result ranking the POIs by the count of GPS points within the POIs could not distinguish these transition events from the user's actual stay at a building. However, since the transition data points are spatially and temporarily sparse, therefore spatio-temporal-density-rank provides much lower rank to the nearby POIs of the transition path. The *Line-count-rank* component is the best performer for both *single place* and *subplace* datasets since it can rank the POIs based on the intersecting line count, even when all the measured GPS positions are outside the POI.

Table 9: Overall NDCG@5 of the components of *POI-ID* and the baseline techniques for our university datasets. A single-place dataset contains data for a user's stay at a single building in the university. A sub-place dataset describes data for a user's stay at multiple nearby buildings.

Methodology	Single place datasets	Sub-place datasets
Distance-to-staypoint (baseline 5)	0.417	0.484
Density-Rank (baseline 6)	0.540	0.65
Point-Count-Rank (baseline 7)	0.559	0.651
Spatio-temporal density rank	0.544	0.675
Line count rank	0.574	0.686

We also performed an experiment to compare the ranking performance of different combinations of *Spatio-temporal-density-rank*, *Point-count-rank* and *Line-count-rank* techniques, since these were the top 3 winners in terms of their overall ranking performance. Table 10 summarizes the average NDCG@5 of all location files for different combinations of the above three techniques. The best performer here is the *POI-ID*, i.e., combining *Spatio-temporal-density-rank* with *line-count-rank*. Significant improvement in ranking performance is obtained for *POI-ID* while comparing against the other combinations (p-value < 0.05).

Table 10: Overall NDCG@5 performance of different combinations of ranking techniques for all 100 datasets. The best performance is obtained by combining Spatio-temporal-density-rank and Line-count-rank as in *POI-ID*

Methodology	NDCG@5
Spatio-temporal-density-rank + Point-count-rank	0.708
Point-count-rank + Line-count-rank	0.725
Spatio-temporal-density-rank + Point-count-rank + Line-count-rank	0.727
Spatio-temporal-density-rank + Line-count-rank (POI-ID)	0.735

4.4.2. Summary of Findings

In summary, the *Spatio-temporal-density-rank* component of *POI-ID* has always better performance in terms of NDCG than the ordinary spatial density rank technique since it also considers the temporal dimension and inaccuracy in density estimation. However, due to high GPS noise in urban regions, sometimes it becomes difficult to distinguish the noise points from the actual data points considering their spatial and temporal closeness. As a result, using the temporal dimension sometimes misdirects the *Spatio-temporal-density-rank* to perform worse than the ordinary *Point-count-rank* technique. On the contrary, the *Spatio-temporal-density-rank* has better performance in determining the spatio-temporarily close events and ranking the POIs corresponding to user's visits to multiple nearby buildings in urban canyons. We also found that the individual performance of *Line-count-rank* component of *POIIdentifier* in terms of NDCG is very good compared to all the baseline approaches. This is because it can rank the POIs even when the user's ground truth location does not contain any GPS data. This technique can easily be extended for real-time scenarios due to its simplicity. The best performance is obtained in terms of both recall and NDCG while combining the two proposed techniques as in *POI-ID*.

5. Conclusion and Future Work

The pervasiveness of GPS enabled mobile devices has created significant volumes of trajectories encapsulating users' movement behaviours. Extracting significant places from these trajectories has attracted considerable research interest due to the large range of applications. Significant places are often indoor locations and may exist in urban canyons where GPS devices can have poor accuracy. Our proposed algorithm, *POI-ID*, is able to estimate a user's actual POI at a building level accuracy from highly noisy GPS data. The experimental results and comparison against the baseline techniques show that *POI-ID* outperforms in terms of both identifying (recall) and ranking (NDCG) a user's true POIs, even for urban regions.

POI-ID can successfully predict a user's POI at building level accuracy. However, for situations like fire-fighting and search and rescue operations, we may need to track users in real-time at a room level accuracy where Wi-Fi, RFID based indoor localization may not be useful anymore. Furthermore, the major challenge of using GPS-enabled mobile devices at indoor places is the large power consumption. In the future we plan to extend our research to investigate these issues.

6. Acknowledgements

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP110100757).

References

[1] M. C. González, C. A. Hidalgo, A.-L. Barabási, Understanding individual human mobility patterns, Nature 458 (7235) (2009) 238–238.

[2] M. A. Saleem, I. Fatima, K. U. Khan, Y. K. Lee, S. Lee, Trajectory based activity monitoring and healthcare provisioning, in: The Tenth IEEE International Conference on Pervasive Intelligence and Computing (PiCom 2012), Changzhou, China (December 2012), 2012.

[3] M. Daubal, O. Fajinmi, L. Jangaard, N. Simonson, B. Yasutake, J. Newell, M. Ali, Safe step: a real-time GPS tracking and analysis system for criminal activities using ankle bracelets, in: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2013, pp. 502–505.

[4] L. Turner, M. Udal, B. Larson, S. Shearer, Monitoring cattle behavior and pasture use with GPS and GIS, Canadian Journal of Animal Science 80 (3) (2000) 405–413.

[5] E. D. Ungar, Z. Henkin, M. Gutman, A. Dolev, A. Genizi, D. Ganskopp, Inference of animal activity from GPS collar data on free-ranging cattle, Rangeland Ecology & Management 58 (3) (2005) 256–266.
[6] R. N. Handcock, D. L. Swain, G. J. Bishop-Hurley, K. P. Patison, T. Wark, P. Valencia, P. Corke, C. J. ONeill, Monitoring animal behaviour and environmental interactions using wireless sensor networks, GPS collars and satellite remote sensing, Sensors 9 (5) (2009) 3586–3603.
[7] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, J. Eriksson, Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2009, pp. 85–98.
[8] J. Yoon, B. Noble, M. Liu, Surface street traffic estimation, in: Proceedings of the 5th international conference on Mobile systems, applications and services, ACM, 2007, pp. 220–232.
[9] P. Mohan, V. N. Padmanabhan, R. Ramjee, Nericell: rich monitoring of road and traffic conditions using mobile smartphones, in: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, 2008, pp. 323–336.
[10] Geolife: Building social networks using human location history, `http://research.microsoft.com/en-us/projects/geolife/`, accessed: 28/01/2014.
[11] foursquare, `https://foursquare.com`, accessed: 28/01/2014.
[12] Y. Zheng, L. Zhang, Z. Ma, X. Xie, W.-Y. Ma, Recommending friends and locations based on individual location history, ACM Transactions on the Web (TWEB) 5 (1) (2011) 5.
[13] V. W. Zheng, Y. Zheng, X. Xie, Q. Yang, Towards mobile intelligence: Learning from GPS history data for collaborative recommendation, Artificial Intelligence 184 (2012) 17–37.
[14] Global Positioning System: Theory & Applications (Volume One) (Progress in Astronautics and Aeronautics), 1st Edition, AIAA (American Institute of Aeronautics & Ast, 1996.
URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/156347106X`
[15] M. B. Kjærgaard, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, K. Grønbæk, Indoor positioning using gps revisited, in: Pervasive Computing, Springer, 2010, pp. 38–56.
[16] P. Misra, P. Enge, Global Positioning System: Signals, Measurements and Performance, revised 2nd Edition, Ganga-Jamuna Press, 2011.
[17] B. Shaw, J. Shea, S. Sinha, A. Hogue, Learning to rank for spatiotemporal search, in: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, WSDM'13, ACM, 2013, pp. 717–726.
[18] K. Laasonen, M. Raento, H. Toivonen, Adaptive on-device location recognition, in: A. Ferscha, F. Mattern (Eds.), Pervasive Computing, Vol. 3001 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 287–304.
[19] J. Krumm, E. Horvitz, Locadio: Inferring motion and location from wi-fi signal strengths, in: in First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Mobiquitous, 2004.
[20] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al., Place lab: Device positioning using radio beacons in the wild, in: Pervasive Computing, Springer, 2005, pp. 116–133.
[21] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, J. Hughes, Learning and recognizing the places we go, in: M. Beigl, S. Intille, J. Rekimoto, H. Tokuda (Eds.), UbiComp 2005: Ubiquitous Computing, Vol. 3660 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 159–176.
[22] J. H. Kang, W. Welbourne, B. Stewart, G. Borriello, Extracting places from traces of locations, in: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, ACM, 2004, pp. 110–118.
[23] M. Kourogi, N. Sakata, T. Okuma, T. Kurata, Indoor/outdoor pedestrian navigation with an embedded gps/rfid/self-contained sensor system, in: Advances in Artificial Reality and Tele-Existence, Springer, 2006, pp. 1310–1321.
[24] N. Eagle, A. (Sandy) Pentland, Reality mining: sensing complex social systems, Personal Ubiquitous Computing. 10 (4) (2006) 255–268.
[25] P. Nurmi, S. Bhattacharya, Identifying meaningful places: The non-parametric way, in: Pervasive Computing, Springer, 2008, pp. 111–127.
[26] D. H. Kim, J. Hightower, R. Govindan, D. Estrin, Discovering semantically meaningful places from pervasive rf-beacons, in: Proceedings of the 11th International Conference on Ubiquitous Computing, Ubicomp '09, ACM, New York, NY, USA, 2009, pp. 21–30.
[27] O. Mezentsev, J. Collin, G. Lachapelle, Pedestrian dead reckoning–a solution to navigation in gps signal degraded areas?, Geomatica 59 (2) (2005) 175–182.
[28] M. DSouza, T. Wark, M. Karunanithi, M. Ros, Evaluation of realtime people tracking for indoor environments using ubiquitous motion sensors and limited wireless network infrastructure, Pervasive and Mobile Computing 9 (4) (2013) 498–515.
[29] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, W. H. Engelmann, et al., The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants, Journal of exposure analysis and environmental epidemiology 11 (3) (2001) 231–252.
[30] D. Ashbrook, T. Starner, Using GPS to learn significant locations and predict movement across multiple users, Personal Ubiquitous Comput. 7 (5) (2003) 275–286.
[31] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, A. Vaisman, A model for enriching trajectories with semantic geographical information, in: Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems, GIS '07, ACM Press, New York, NY, USA, 2007.
[32] A. T. Palma, V. Bogorny, B. Kuijpers, L. O. Alvares, A clustering-based approach for discovering interesting places in trajectories, in: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08, ACM Press, New York, NY, USA, 2008, pp. 863–868.
[33] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM Press, New York, NY, USA, 2009, pp. 791–800.
[34] V. W. Zheng, Y. Zheng, X. Xie, Q. Yang, Collaborative location and activity recommendations with GPS history data, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, USA, 2010, pp. 1029–1038.
[35] X. Cao, G. Cong, C. S. Jensen, Mining significant semantic locations from GPS data, Proceedings of the VLDB Endowment 3 (1-2) (2010) 1009–1020.
[36] T. Bhattacharya, L. Kulik, J. Bailey, Extracting significant places from mobile user GPS trajectories: A bearing change based approach, in: Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '12, 2012.

[37] Google+, https://play.google.com/store/apps/details?id=com.google.android.apps.plus, accessed: 28/01/2014.
[38] facebook: Share where you are, https://www.facebook.com/about/location, accessed: 28/01/2014.
[39] Instagram: Capture and share the world's moments, http://instagram.com, accessed: 28/01/2014.
[40] R. G. Congalton, K. Green, Assessing the accuracy of remotely sensed data: principles and practices, CRC press, 2008.
[41] B. J. Arnow, On laplace's extension of the buffon needle problem, The College Mathematics Journal 25 (1) (1994) 40–43.
[42] M. Ester, H.-p. Kriegel, J. Sander, X. Xu, A density–based algorithm for discovering clusters in large spatial databases with noise, Computer 1996 (6) (1996) 226–231.
[43] N. Marmasse, C. Schmandt, Location-aware information delivery with commotion, in: Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing, HUC '00, Springer-Verlag, London, UK, UK, 2000, pp. 157–171.
[44] R. Hariharan, K. Toyama, Project lachesis: parsing and modeling location histories, in: Geographic Information Science, Springer, 2004, pp. 106–124.
[45] X. Xiao, Y. Zheng, Q. Luo, X. Xie, Finding similar users using category-based location history, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, ACM, New York, NY, USA, 2010, pp. 442–445.
[46] L. Liao, D. Fox, H. Kautz, Extracting places and activities from gps traces using hierarchical conditional random fields, The International Journal of Robotics Research 26 (1) (2007) 119–134.
[47] C. Lee, G. Yoon, D. Han, A probabilistic place extraction algorithm based on a superstate model, IEEE Transactions on Mobile Computing (2013) 945–956.
[48] D. H. Kim, Y. Kim, D. Estrin, M. B. Srivastava, Sensloc: sensing everyday places and paths using less energy, in: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ACM, 2010, pp. 43–56.
[49] N. Brouwers, M. Woehrle, Dwelling in the canyons: Dwelling detection in urban environments using GPS, wi-fi, and geolocation, Pervasive and Mobile Computing (Special issue on Pervasive Urban Applications) 9 (5) (2013) 665–680.
[50] N. D. Lane, D. Lymberopoulos, F. Zhao, A. T. Campbell, Hapori: context-based local search for mobile phones using community behavioral modeling and similarity, in: Proceedings of the 12th ACM international conference on Ubiquitous computing, ACM, 2010, pp. 109–118.
[51] D. Lian, X. Xie, Learning location naming from user check-in histories, in: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2011, pp. 112–121.
[52] I. Road Engineering Journal, TranSafety, Study compares older and younger pedestrian walking speeds, http://www.usroads.com/journals/p/rej/9710/re971001.htm, accessed: 20/11/2013.
[53] H. J. Ralston, Energy-speed relation and optimal speed during level walking, Internationale Zeitschrift für angewandte Physiologie einschließlich Arbeitsphysiologie 17 (4) (1958) 277–283.
[54] R. McNeill Alexander, Energetics and optimization of human walking and running: the 2000 raymond pearl memorial lecture, American Journal of Human Biology 14 (5) (2002) 641–648.
[55] J. E. Bertram, A. Ruina, Multiple walking speed–frequency relations are predicted by constrained optimization, Journal of theoretical Biology 209 (4) (2001) 445–453.
[56] J. E. Bertram, Constrained optimization in human walking: cost minimization and gait plasticity, Journal of experimental biology 208 (6) (2005) 979–991.
[57] D. W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics), 1st Edition, Wiley, 1992.
[58] R. V. Hogg, E. A. Tanis, Probability and Statistical Inference, 5th Edition, Prentice Hall, 1997.
[59] P. Mathews, The circular normal distribution, http://www.mmbstatistical.com/ToT/distofr.pdf (2000).
[60] T. Duong, M. Hazelton, Plug-in bandwidth matrices for bivariate kernel density estimation, Journal of Nonparametric Statistics 15 (1) (2003) 17–30.
[61] T. Duong, ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r, Journal of Statistical Software 21 (7) (2007) 1–16.
[62] T. Duong, Bandwidth selectors for multivariate kernel density estimation, Ph.D. thesis, School of Mathematics and Statistics, University of Western Australia, Australia (October 2004).
[63] M. Rudemo, Empirical choice of histograms and kernel density estimators, Scandinavian Journal of Statistics (1982) 65–78.
[64] A. W. Bowman, An alternative method of cross-validation for the smoothing of density estimates, Biometrika 71 (2) (1984) 353–360.
[65] T. Duong, M. L. Hazelton, Cross-validation bandwidth matrices for multivariate kernel density estimation, Scandinavian Journal of Statistics 32 (3) (2005) 485–506.
[66] J. Chacón, T. Duong, Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices, Test 19 (2) (2010) 375–398.
[67] J. E. Chacón, T. Duong, Unconstrained pilot selectors for smoothed cross-validation, Australian & New Zealand Journal of Statistics 53 (3) (2011) 331–351.
[68] J. Ramaley, Buffon's noodle problem, The American Mathematical Monthly 76 (8) (1969) 916–918.
[69] E. M. Voorhees, The TREC-8 question answering track report., in: TREC, Vol. 99, 1999, pp. 77–82.
[70] Geolife GPS trajectories, http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/, accessed: 03/10/2013.
[71] GPS share, http://www.gpsshare.com, accessed: 03/10/2013.
[72] Bikely, http://www.bikely.com, accessed: 03/10/2013.
[73] Google maps api v2 geocoder tool, http://gmaps-samples.googlecode.com/svn/trunk/geocoder/v2-geocoder-tool.html, accessed: 03/10/2013.
[74] Openstreetmap. the free wiki world map, http://www.openstreetmap.org, accessed: 03/10/2013.
[75] Y. Zheng, X. E. Zhou, Computing with Spatial Trajectories, 1st Edition, Springer, 2011.
[76] K. Järvelin, J. Kekäläinen, IR evaluation methods for retrieving highly relevant documents, in: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2000, pp. 41–48.