Extracting Hotspots without A-priori by Enabling Signal Processing over Geospatial Data

Vaibhav Kulkarni, Arielle Moro, Bertil Chapuis, Benoît Garbinato

{firstname.lastname}@unil.ch
Distributed Object Programming Laboratory
University of Lausanne, Switzerland

ABSTRACT

The proliferation of mobile devices equipped with internet connectivity and global positioning functionality (GPS) has resulted in the generation of large volumes of spatiotemporal data. This has led to the rapid evolution of location-based services. The anticipatory nature of these services, demand exploitation of a broader range of user information for service personalization. Determining the users' places of interest, i.e. *hotspots* is critical to understand their behaviors and preferences. Existing techniques to detect hotspots rely on a set of *a-priori* determined parameters that are either dataset dependent or derived without any empirical basis. This leads to biased results and inaccuracies in estimating the total number of hotspots belonging to a user, their shape and the average dwelling time. In this paper, we propose a parameter-less technique for extracting hotspots from spatiotemporal trajectories without any *a-priori* assumptions. We eliminate parameter dependence by treating trajectories as spatiotemporal signals and rely on signal processing algorithms to derive hotspots. We experimentally show that, our technique does not necessitate any spatiotemporal or behavior dependent bounds, which makes it suitable to extract hotspots from a larger variety of datasets and across users having disparate mobility behaviors. Our evaluation results on a real world dataset, show accuracy rates exceeding 80% and outperforms traditional clustering techniques used for hotspot detection.

CCS Concepts

•Information systems → Spatial-temporal systems; Location based services;

Keywords

Spatiotemporal hotspots; Clustering parameters; Signal processing

1. INTRODUCTION

An integral aspect of location-based services (LBS) is to extract meaningful information from the location trajectories recorded by their users. For example, mobility prediction services rely on clustering algorithms to extract user specific points of interest from raw GPS trajectories. LBS typically depend on mobility prediction as a means to improve quality of service by pushing context-aware information to users ahead of time. Other services such as traffic management, urban planning and consumer profiling, heavily rely on their ability to identify the hotspots where moving entities spend a considerable amount of time. Hotspot detection is therefore a key aspect of LBS for user mobility modeling.

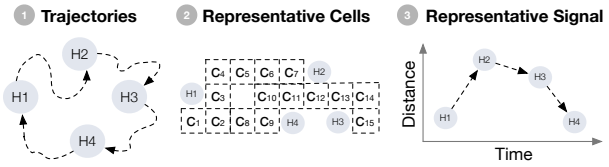

Figure 1: From 3-D traces to 2-D signals

Several techniques for extracting hotspots from trajectories were inspired by well-known clustering algorithms such as k-means [6] and DBSCAN [2]. Other methods span the domain of scan statistics, fingerprinting, gradient-based or eigenvector-tensor based techniques [3, 12, 13]. Overall, these techniques rely on a set of parameters that reflect *a-priori* assumptions about the mobility behavior of users by imposing bounds on distance, speed, time, number of points and/or visitation repeatability rates. These techniques require multiple steps, involving several iterations over the dataset to extract all the hotspots resulting in an increased latency. Furthermore, the hotspots are assumed to fit a predefined shape (mostly circular) which rarely reflects the reality, leading to erroneous estimations of hotspot area and dwelling time.

To address these problems, we propose a hotspot detection technique that is independent of such *a-priori* assumptions. We treat user mobility trajectories as spatiotemporal signals (see Figure 1) and apply filtering modules to iteratively extract and enhance the quality of the detected hotspots. We show that, these signals preserve all the key knowledge contained in the trajectories, and our system is able to accurately detect the hotspot occurrences, the time of hotspot entry and exit and a precise representation of the total area and time spent at each hotspot. We evaluate the extracted hotspots in terms of precision and recall rates and compare its efficacy with respect to popular clustering techniques used for hotspot detection. Applying signal processing algorithms also has the added benefit of exploiting the digital-signal processors (DSP) embedded in modern smartphones. This in turn preserves the privacy of users by restricting the computations on their smartphones by not sharing the raw data with untrusted third-party services.

The rest of the paper is structured as follows. The related work on hotspot detection and the associated drawbacks are presented in Section 2. We present the problem statement addressed in this paper and the translation process from trajectories to signals in Section 3

SIGSPATIAL'17 November 7–10, 2017, Los Angeles Area, CA, USA

ACM ISBN 978-1-4503-5490-5/17/11.

DOI: https://doi.org/10.1145/3139958.3140002

(a) A typical user trajectory. The green circles indicate user hotspots (b) The trajectory after being projected on to the google S2 cells (c) The signal generated after plotting the cell IDs' against timestamps

Figure 2: Translating geospatial trajectories to space-time signals (left to right)

and Section 4 respectively. The system design and implementation is described in Section 5. The evaluation results and discussion is presented in Section 6. We finally conclude the paper in Section 7.

2. RELATED WORK

In this section, we review the existing techniques to mine hotspots from spatiotemporal data. The premier contribution in adopting the data clustering techniques for hotspot detection was made by Ashbrook et al. in [1]. They propose an iterative approach to extract the hotspots and improve their granularity by imposing spatiotemporal bounds. These bounds are derived by analyzing their variance with respect to certain values they affect. Montoliu et al. proposed a two-level clustering approach in [10], where the geolocated points are first clustered in the temporal domain to discover the stay points that are used to derive stay regions using a grid-based clustering approach. Several other popular clustering algorithms such as Density-Joinable clustering [15], Density-Time clustering [6] and Time-Density clustering [5] have also been adapted to detect clusters in geospatial datasets, which are then considered as hotspots. Detecting hotspots by leveraging realtime location data streams has been proposed in [11, 8]. This scheme extracts the most frequent and recent hotspots of users in realtime. Zheng et al. [14] propose a variation of DBSCAN, wherein the input parameters are estimated by observing the distribution of movement density. Thomason et al. [12] proposed a gradient-based technique that combines the advantages of both; k-means and DBSCAN.

Farrahi et al. proposes a fingerprinting-based approach [4] by analyzing the temporal regularities and patterns of local transitions over time. The fingerprint is essentially a vector of visible cell towers, as described in the works of BeaconPrint [7] and the hotspots are detected based on the repeatability rate of an associated vector. Another set of techniques are based on scan-statistics, wherein a cylinder of varying radii and height is moved over the spatiotemporal space. The surface of the cylinder covers the space dimension and its height covers the time dimension. The cylinders are then sorted depending on a parameter called *p-value*, which is then used as a threshold to consider the detected regions as hotspots. Louail et al. propose a technique to extract hotspots from trajectories belonging to a group of users without relying on the commonly used spatiotemporal bounds [9]. However, they consider a group of geolocated points at a particular time t, as a hotspot, if the density of users at that location is greater then a predefined threshold δ.

The above techniques use several bounds to classify a particular region as a hotspot. Some of the parameters include, maximum distance between the collected locations, maximum and minimum time bound, cluster shape, grid size and maximum number of points per cluster. However, different users are characterized by different mobility profiles, which result in varying optimal values of these *a-priori* chosen parameters. This task of estimating the parameters and their values is challenging due to the inherent number of possible parameter combinations, different mobility behaviors, duration of the available dataset, rate at which locations are sampled and the distribution of noise in the recorded data. Often, the parameter values are derived based on logical reasoning and lack exhaustive empirical basis. This could result in a possible bias when generalizing and comparing results obtained with different techniques on different datasets. This has resulted in conflicting views regarding the significance of some parameters, such as maximum time bound between two coordinate points as seen in [10] and [12]. In this paper, we propose a solution to address the above discussed problems.

3. PROBLEM STATEMENT

Our work identifies and addresses the problem of extracting hotspots from user location trajectories without relying on any rigid parameters. Our solution to address the problem is to treat user mobility trajectories as space-time signals and process these signals to extract the hotspots. We thus have a two-fold problem statement as described below:

1. Given a user's trajectory $T = \langle ..., t_i, ... \rangle$, a sequence of spatiotemporal points, where each point t_i is a three item tuple, $\langle lat_i, lon_i, t_i \rangle$, where lat and lon is the latitude-longitude coordinate pair and t is the timestamp, translate it into a 2-dimensional continuous signal, $s(t)$, modeled as a function of changing distance with respect to time, retaining the spatial locality between the discretized points.
2. Given the spatiotemporal signal $s(t)$ of a user u, extract all the distinct hotspots and their properties, namely; the area and dwelling time.

4. FROM TRAJECTORIES TO SIGNALS

The geolocation sensors result in noise and non-uniformly sampled data points due to the hardware imperfections and network failures. Thus, the data points need to be de-noised and resampled to generate a continuous signal. We first use a standard convolution-based low-pass filter to remove the noisy components residing at high frequencies. Next, in order to get a uniformly sampled location points we apply a semivariance interpolation scheme, which fits the missing points by modeling the similarity between the points as a function of changing distance [8].

The output of the preprocessing stage is a uniformly sampled and de-noised coordinate points. In order to discretize space, we use the Google S2 library[1]. The library projects a spatial region on to the face of a cube which encloses the sphere. It performs a hierarchical decomposition of the sphere into compact cells and superimposes the spatial region/point on to the cells. It then constructs a quad-tree on each face and selects a quad-tree cell containing the projected region. Each cell is represented by exactly the same area and provides sufficient resolution for indexing the geographic features. The cells are enumerated on the Hilbert curve, which preserves the spatial locality of the points. The resulting spatiotemporal signal can be denoted as $s(t) = \langle ..., (c_i, t_i), ... \rangle$, where c_i is the cell ID and t_i the timestamp as shown in Figure 2.

[1]Google S2: https://pypi.python.org/pypi/s2sphere/

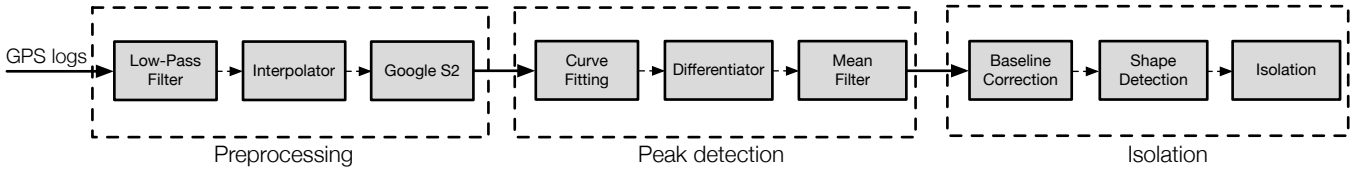

Figure 3: From preprocessing to peak detection and hotspot extraction

5. SYSTEM DESIGN

In this section, we present our system design and implementation. The mobility signal $s(t)$ has two main components: (1) a static element which corresponds to the place having maximum user time occupancy, (2) local maxima/minima, which correlate with the user's frequently visited places. The user movements oscillate around the static element with a significant deviation, generating several local maxima/minima. This can be viewed as the presence of basic noise with a general mean which makes the hotspots identifiable . Therefore, the problem of hotspot discovery is essentially a matter of detecting the local maxima and minima (or simply 'peaks') contained in the signal. In order to determine the hotspot properties, we design our system to heuristically compute the peak start and end positions, peak height and width to estimate the hotspot visit entry and end time, the total distance travelled and the total area. The steps involved in hotspot detection are illustrated in Figure 3.

In order to make the peaks distinct, we perform curve fitting over the discretized location traces, $s(t)$. However, the peak shapes are not identical throughout the signal and differ according to the visited place. We therefore perform a non-linear iterative curve fitting, ensuring that the peaks do not shift or are missed. The peaks can then be detected by taking the first differential of the curves. The detection procedure operates by checking for the point of downward/upward going zero-crossing at the peak-maximum/minimum, for the peaks and the valleys. In order to make the peak detection robust, we apply a mean filter and smooth the first differential prior to checking the upward/downward-going zero-crossings. The detected peaks in turn contain two components, the travel path to the hotspot and the hotspot region itself. Thus, in the next step we split the ravel and the hotspot component in the peak, which requires a correct estimation of the peak shape. This step requires automatic adjustment of the baselines so as to adapt constantly to the changing user behaviors. To address this, we keep a track of the standard deviation of the points and analyze the points deviating from the moving mean, which iteratively sets the baseline and operates precisely irrespective of the peak shape. The peak shape is finally detected by taking the successive derivatives, as different peak shapes have distinct derivative shapes.

Finally, for isolating the peak components, we monitor the average rate of change of slope of the detected peaks. Once a user arrives at the hotspot, the slope changes to zero or to an infinitesimally small value, as compared to the slope of the travel component. Thus the two parts can be separated depending on the average rate of change of the slope along the maxima or minima. After the peak is isolated, the cells belonging to the hotspot are extracted and the remaining cells belong to the travel component. The representative path connecting the hotspot is constructed by selecting the cells common to both the edges. The cells, when inverted back to the location coordinates, represent the spatial locations.

6. EVALUATION AND DISCUSSION

In order to evaluate the accuracy of the detected hotspots, we validate our results with the ground truth and perform a comparison with three popular clustering techniques commonly used for hotspot detection. As the publicly available datasets are devoid of the ground truth, we collect a dataset to validate the efficacy of our approach and to confirm our findings regarding the correlation between the spatiotemporal components and the signal elements. The mobile application provided to the users logs their latitude, longitude, timestamp, acceleration, altitude, horizontal and vertical accuracy of the GPS coordinates for a period of 11 weeks. The data points are collected at a sampling rate of 5 seconds with a granularity of resolution up to 5 meters. The ground truth is captured by periodically attesting the visited hotspots. The hotspots were selected with a clear definition: *'any place where the subject visited with an intentional purpose'*. These regions include places such as cafeterias, restaurants, bus/train/metro stops, sports arenas, bookstores, office and work places and excursions. The ground-truth evaluation was performed by computing the precision, recall and the accuracy values.

We configure the Google S2 library to map each coordinate pair to a cell of dimension $38m^2$. It could be argued that the cell size involves a arbitrarily chosen parameter in the process. However, real-world hotspots typically spread over areas larger than $38m^2$. Furthermore, this choice is motivated by the localization accuracy of a typical GPS sensor and the performance complexity involved when subdividing the cells to the leaf level.

We consider Density Joinable Cluster (DJ Cluster) [15], Density Time Cluster (DT Cluster) [6] and ZOI Detect [8]. DJ Cluster computes hotspots, based on the number of points within a certain radius and merges these clusters if they share at least one point in common. Furthermore, the points are also clustered if they satisfy the min_{speed} bound. DT Cluster aggregates points lying within a predetermined spatiotemporal bound. These clusters are treated as valid hotspots. ZOI Detect follows a similar strategy as DT cluster but involves an additional parameter min_{visit} as a threshold and merges the clusters upon intersection. The parameters selected by these techniques and their values are shown in Table 1. These values selected in published works are either based on dataset trends [8] or on user mobility behavior [15].

Clustering algorithm	Parameters
DJ Cluster	Min_{speed}: 0.4 (km/hour) / $Cluster_{radius}$: 60.0 (meters) / Min_{points}: 10
DT Cluster	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds)
ZOI Detect	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds) / Min_{visit}: 6

Table 1: Clustering algorithms and their default parameter values

We see that DT Cluster and ZOI Detect have a high precision and low recall and accuracy rate as seen in Figure 4. This indicates that, these techniques discover a large number of hotspots that are not contained in the true hotspot set. This is clearly due to the spatiotemporal bounds being too rigid, which results in considering arbitrary clusters as valid hotspots. DJ cluster, however, has higher recall and low precision. Here, we see that the Min_{speed} eliminates the occurrences of false negatives, whereas, the Min_{points} creates high false positives. Increasing the Min_{points} can address such occurrences, as it requires a higher density of points, thus creating only valid hotspots. In case of our method, we have a few false positives due to the high sensitivity for the slop change and only

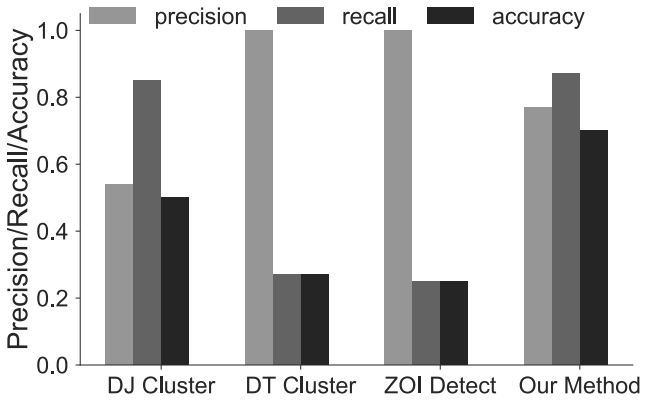

Figure 4: Ground truth validation

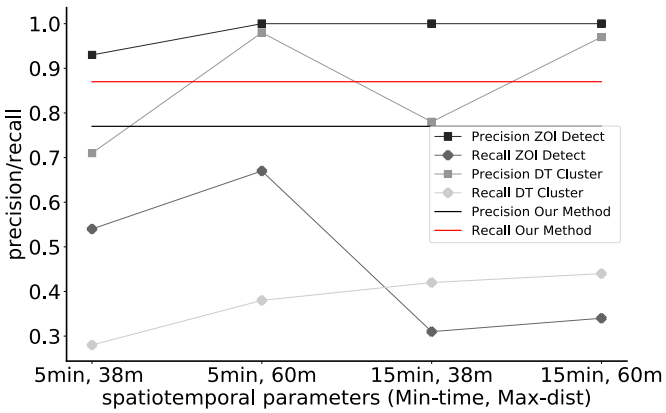

Figure 5: Impact of the parameters on the hotspot accuracy

three false negatives. Closely examining the false-positives reveal that, these regions are visits without any purpose attached, such as delays at metro and bus stops. This creates additional hotspots which are not based on user intent. The false negatives are the stops where the user does not have to wait. These cases occur due to planned time synchronization by the user between the transportation mode switches, resulting in a constant average slope.

To better understand the parameter influence, we consider four different parameter sets for the values of Min_{time} and Max_{dist} as seen in Figure 5. We see that the parameter Min_{visits} always correctly classifies a region as a hotspot, thus leading to high precision rates. We can also see that larger values of Max_{dist} results in higher precision and recall in DT Cluster. Max_{dist}, thus plays a vital role in determining precision, compared to Min_{time} parameter in the considered dataset. These results highlight the importance of selecting the parameter space which is challenging to determine *a–priori*.

In general, we observe that DJ Cluster and DT Cluster detect a significantly high number of hotspots in both the cases. In case of DJ Cluster, we find that the parameter Min_{points} creates a large number of hotspots. However, we argue that if the sampling rate of the dataset is high, Min_{speed} could play an important role in further increasing the clusters. Whereas, in DT Cluster Min_{time} bound parameter results in a higher frequency of visit separations, increasing the total number of hotspots. These factors contribute to a higher number of hotspots, which is not typical for an average user. We observe that the number of hotspots discovered by ZOI Detect [11, 8] is lower than DT and DJ Cluster. This is due to the merging of individual clusters upon intersection, in addition to extracting the most frequent clusters governed by the Min_{visit} parameter. In general, if the parameters satisfy cluster merging, multiple clusters merge and form a large hotspot; hotspot division occurs if this bound is missed by even an infinitesimal small value. This results in the fluctuation of the number of hotspots solely due to the parameters.

The hotspot area in our case corresponds to $38m^2 \times cellnumbers$. We find that, this results in a significantly smaller areas compared to the clustering techniques and overlaps with the ground truth area. This is due to the set of cells in the hotspot, which corresponds to a cell where a user was actually present, and which is smaller than the actual area of the hotspot.

7. CONCLUSION

In this paper, we have proposed a technique to detect hotspots from user trajectories without relying on any *a-priori* assumptions. We have depicted the bias resulting due to the stringent parameter bounds while extracting user hotspots. We have also depicted the problems arising from such bounds that are based on non-empirical calculations and extended to operate on some other datasets and on users having different mobility behaviors. We have addressed this problem by treating user movements as spatiotemporal signals, effectively converting it to a peak-detection problem by using signal-processing algorithms. The evaluation results show that our approach outperforms the popular clustering techniques used for hotspot detection. We have also validated our results with the ground truth and achieved precision and recall rates exceeding 80%.

Acknowledgment. This work is partially supported by the Swiss National Science Foundation grant 146714 and 157160.

8. REFERENCES

[1] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps. In *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pages 101–108. IEEE, 2002.

[2] M. Ester, H.-p. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[3] H. Fanaee-T and J. Gama. An eigenvector-based hotspot detection. *arXiv preprint arXiv:1406.3191*, 2014.

[4] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):3, 2011.

[5] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, pages 103–126, 2011.

[6] R. Hariharan and K. Toyama. Project lachesis: parsing and modeling location histories. In *International Conference on Geographic Information Science*, pages 106–124. Springer, 2004.

[7] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *International Conference on Ubiquitous Computing*, pages 159–176. Springer, 2005.

[8] V. Kulkarni, A. Moro, and B. Garbinato. Mobidict: A mobility prediction system leveraging realtime location data streams. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, IWGS '16, pages 8:1–8:10, New York, NY, USA, 2016. ACM.

[9] T. Louail, M. Lenormand, O. G. C. Ros, M. Picornell, R. Herranz, E. Frias-Martinez, J. J. Ramasco, and M. Barthelemy. From mobile phone data to the spatial structure of cities. In *Scientific reports*, 2014.

[10] R. Montoliu and D. Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[11] A. Moro and B. Garbinato. A system-level architecture for fine-grained privacy control in location-based services. In *2016 12th European Dependable Computing Conference (EDCC)*, pages 25–36, Sept 2016.

[12] A. Thomason, N. Griffiths, and V. Sanchez. Identifying locations from geospatial trajectories. *Journal of Computer and System Sciences*, 82(4):566–581, 2016.

[13] Q. Zhao, Y. Shi, Q. Liu, and P. Fränti. A grid-growing clustering algorithm for geo-spatial data. *Pattern Recognition Letters*, 53:77–84, 2015.

[14] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[15] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 266–273. ACM, 2004.